Pushing the Limits of Indoor Localization in Today's Wi-Fi Networks

JIE XIONG

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of the

University College London.

Department of Computer Science

University College London

Aug 2015

I, JIE XIONG, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

Wi-Fi networks are ubiquitous nowadays and play an increasingly important role in our everyday lives. Many emerging applications including augmented reality, indoor navigation and human tracking, rely heavily on Wi-Fi. One key component for the success of these applications is accurate localization. While we have GPS in the outdoor environment, indoor localization at a sub-meter granularity remains challenging.

On the other hand, Wi-Fi technology has developed significantly evolving from 802.11b/a/g to the latest 802.11ac standard. In Wi-Fi's development, one interesting trend is the increasing number of antennas attached to a single access point (AP). Another trend is the presence of frequency-agile radios and larger bandwidths in the latest 802.11n/ac standards. These opportunities are leveraged to increase the accuracy of indoor localization significantly:

ArrayTrack employs multi-antenna APs for angle-of-arrival (AoA) information to localize clients accurately indoors. It is the first indoor Wi-Fi localization system able to achieve below half meter median accuracy. Innovative multipath identification scheme is proposed to handle the challenging multipath issue in indoor environment. ArrayTrack is robust in term of signal to noise ratio, collision and device orientation. ArrayTrack does not require any offline training and the computational load is small, making it a great candidate for real-time services.

ToneTrack is a fine-grained indoor localization system employing time difference of arrival scheme (TDoA). ToneTrack uses a novel channel combination algorithm to increase the effective bandwidth without increasing the radio's sampling rate, for higher-resolution time information. A new spectrum identification scheme is proposed to retrieve useful information from a ToA profile even when the overall profile is mostly inaccurate. The triangle inequality property is applied to detect the APs whose direct path is 100% blocked. With a combination of only three 20 MHz channels, ToneTrack achieves below one meter median error, outperforming the traditional super-resolution ToA schemes.

Acknowledgements

I would like to give special thanks to my advisor Dr. Kyle Jamieson for his guidance and advice throughout my research. Prof. Brad Karp and Prof. Mark Handley also gave precious advice to my work throughput the process. My thanks also go to the examiners of my Ph.D thesis, Prof. Jon Crowcroft and Prof. Kai-Kit Wong for their time and valuable feedback. Finally, thanks and love go to my parents and my fiancée for their continuous supports.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the past few years, wireless data connectivity has continued its transition into an essential utility. Wireless networks are now deployed everywhere including enterprises, campuses, airports, homes and most of the shops. On the other hand, almost all mobile devices now are integrated with Wi-Fi chipsets, allowing users easy network connectivity at hotspots. Cellular phones are also ubiquitous, with the steady progression in cellular standards driving data rates increasingly higher to meet an increasing demand. Today's worldwide wireless networking industry is a trillion-dollar business and an indispensable part of the global economy.

On the other hand, Wi-Fi technology is still relatively young. In 1999, the IEEE 802.11b [39] standard was drafted to provide a throughput up to 11 Mbit/s on the 2.4 GHz band. Since then, Wi-Fi technology has developed significantly, evolving from 802.11b to 802.11a/g [41, 40] quickly in the next few years. OFDM (orthogonal frequency division multiplexing) was introduced in 802.11a on the 5 GHz band to increase the throughput. 802.11g employs the same OFDM modulation scheme to provide a maximum raw data rate of 54 Mbit/s at 2.4 GHz. The 802.11n [42] standard was initiated in January 2004 and finalized in 2008 with its formal publication in 2009. 802.11n can provide much better throughput performance and keep pace with the rapidly growing speeds provided by the Ethernet. In order to achieve this higher throughput, quite a few new features have been incorporated into 802.11n. The key feature is the use of Multiple-Input and Multiple-Output (MIMO) technology. MIMO is a technique that exploits multipath propagation opportunities to improve diversity and capacity. When a signal is transmitted from the access point (AP) A to a client B, the signals reach the receiving antennas via multiple paths. The data are split into a number of *spatial streams* and transmitted through separate antennas to the receiver. The current 802.11n standard supports up to four spatial streams which means four antennas are attached to a single 802.11n access point. The MIMO scheme included in 802.11n is still single-user MIMO which means the multiple streams can be sent to only one client (the client may have multiple antennas).

802.11ac [43] is the latest Wi-Fi standard, finalized in 2014. It incorporates multi-user MIMO scheme into the standard which is a big step forward compared to 802.11n. The key feature of 802.11ac is the shift from a single-client to a multi-client communication pattern. Previously the AP can only transmit to one client at a time while 802.11ac enables the AP to transmit to several clients simultaneously. 802.11ac supports up to eight simultaneous spatial streams to four different users. The bandwidth supported is also increased up to 160 MHz, which is four times the bandwidth supported by 802.11n. The latest 802.11ac standard offers clients near an access point (AP) a maximum transmission rate of 1300 Mbit/s, more than enough to stream high definition TV content.

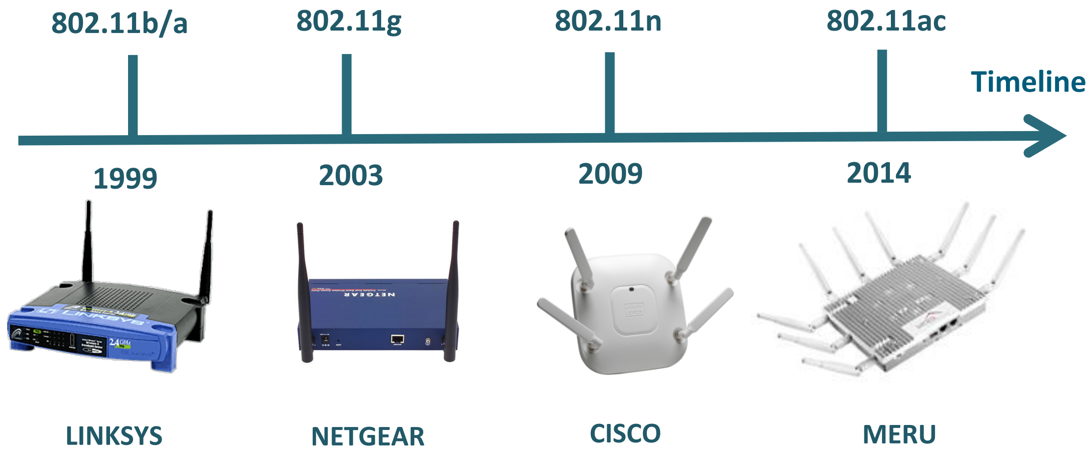


Figure 1.1: Commercial 802.11n/ac APs with multiple antennas.

With the popularity of MIMO and standardization of 802.11n and 802.11ac, one interesting observation is that more and more antennas are attached to a single AP. A lot of commercial multi-antenna APs from major vendors are already on the market as shown in Figure 1.1. 16 sectored antenna APs are available on the market while large-scale MU-MIMO APs with tens of antennas are available in university labs, as shown in Figure 1.2. Also the massive MIMO concept will probably be a key feature included in next generation, so-called 5G cellular networks.

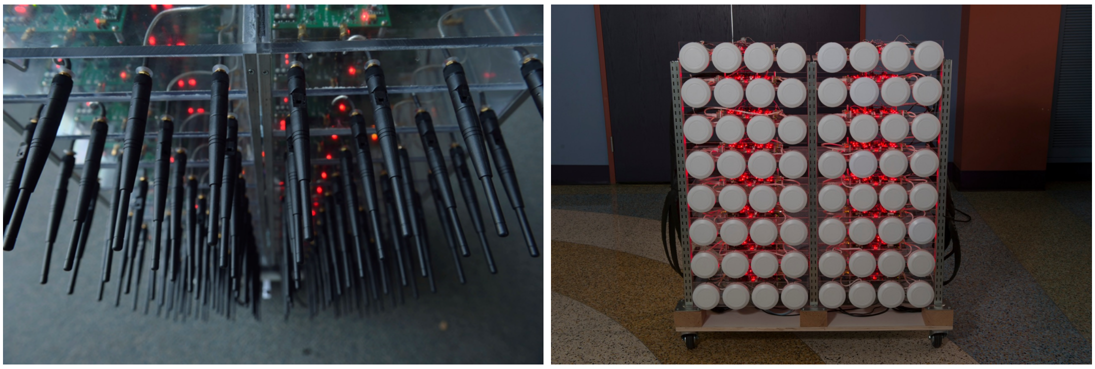


Figure 1.2: Massive MIMO APs from Rice Argos project [1] with many antennas.

In the last 10 years, mobile computing has greatly changed our lives. Now wearables, drones, augmented reality devices such as Microsoft Hololens and even driverless cars have all arrived on the scene. These will for sure change our lives even more in the next decade. For the success of these applications, one critical factor is accurate localization. The Google driverless cars employ a map with error in inches [33] because the distances between the cars can be reduced to a few inches. The wearables need the users' precise localizations in the shopping mall to deliver the most related and useful coupons. Several meters of error can locate a user from one shop to another. In a huge library or supermarket, a lot of people may have experienced difficulty in locating a particular item. Indoor navigation service with centimeter level accuracy will be a very useful application on your mobiles in these scenarios.

In the outdoor environment, GPS is very popular and performs well for most applications. However, in the indoor environment, we still lack an accurate localization system. GPS signals fade significantly in the indoor environment making them too weak to be employed for indoor localization purposes. The granularity level of GPS on the scale of tens of meters are also not enough for a lot of indoor applications which requires meter and even sub-meter level accuracy.

Many technologies have been employed to provide accurate indoor localization services including inferred, ultrasound, camera, sound, light, RFID and Wi-Fi. Among these technologies, Wi-Fi is most promising because of its ubiquitousness. Wi-Fi nowadays is deployed more or less everywhere. Employing other technologies such as ultrasound and camera provides us satisfactory localization accuracy but incurs extra efforts of infrastructure deployment at high cost. So this thesis investigates hosting indoor localization systems on top of the existing Wi-Fi infrastructure.

Among the Wi-Fi based indoor location schemes, there are three main categories: RSSI (received signal strength indicator), AoA (angle of arrival) and ToA/TDoA (time of arrival/time difference of arrival). The pioneer work for RSSI based scheme is RADAR [6] proposed in 1999 which is able to provide a 2-3 meter accuracy. Due to strong multipath in the indoor environment, RSSI based schemes suffer from low accuracy and heavy calibration/training load. A lot of works are proposed to either improve the accuracy or reduce the calibration load. The AoA based localization scheme was popular in radar system but not explored much in Wi-Fi domain mainly because the limited number of antennas on the wireless AP. On the other hand, the accuracy level of ToA/TDoA based indoor localization was restricted by the bandwidth of the channel. With a 20 MHz narrow band channel commonly used in 802.11, we can only achieve a very coarse accuracy, not satisfying the requirement of a lot of applications.

In this thesis, the unique opportunities of current Wi-Fi networks are employed to push the limits of indoor localization in both AoA and ToA domains. With more antennas attached to a single AP with 802.11n and 802.11ac, AoA scheme is employed to provide fine-grained indoor localization accuracy below half meter. Leveraging the frequency-agile radios, a novel channel combination scheme that combines small bandwidth channels is proposed to form a virtual larger bandwidth channel to significantly improve the accuracy for ToA/TDoA based localization. The two indoor localization systems proposed are briefly described below :

ArrayTrack is a realtime indoor localization system which employs an AoA approach for localization. With multiple antennas on a single AP, AoA information is generated when the client's transmission is received at the AP. The location estimate is obtained with AoA information from several such multi-antenna APs. It is first implemented on a software-defined radio platform (Rice WARP [85]) and later on cheap off-the-shelf commodity hardware, where both demonstrate significant performance improvement over the existing systems. ArrayTrack is a robust real-time localization system achieving a 23 cm median accuracy with six APs and eight antennas attached to each AP. The most challenging part of AoA localization is the presence of strong multipath propagation indoors, which leads to performance degradation as only the direct path is pointing to the true location of a client. A novel multipath identification scheme is proposed to tackle this problem. It is observed that the line-of-sight (LOS) path is much more stable compared to the reflection paths on the AoA spectrum when the client incurs a small movement. When a phone is held in hand or pocket, natural human movements present a unique opportunity to identify the reflection paths and improve localization accuracy by taking several measurements and identifying the stable path on the AoA profiles.

ToneTrack is a TDoA-based indoor localization system. TDoA scheme was previously not popular for indoor localization because 802.11 APs do not usually have a large enough bandwidth for necessary time-domain resolution (2 m). A novel channel combination scheme is proposed to form a larger virtual bandwidth which is able to achieve finer resolution in time domain to differentiate direct path and reflection paths. A spectrum identification scheme is proposed to identify the useful part and still retrieve it from an overall inaccurate ToA profile. ToneTrack also employs the triangle inequality property to effectively handle the challenging scenario when a line-of-sight path is 100% blocked. With all these novel schemes applied, ToneTrack is able to achieve below one meter median accuracy with only three transmissions from adjacent 20 MHz channels. ToneTrack can be applied on the 5 GHz and 60 GHz bands to achieve millimeter level accuracy.

The rest of this thesis is structured as follows. Following a literature review in Chapter 2, I present ArrayTrack in Chapter 3 and ToneTrack in Chapter 4 respectively. Chapter 6 concludes the report and Chapter 5 discusses the future work.

Chapter 2

Literature Review

In this chapter, the IEEE 802.11 standards are introduced in chronological order followed by the MIMO technologies which have a close relationship with indoor wireless localization systems. After that, the representative indoor location systems proposed in the last 20 years are discussed in detail.

2.1 IEEE 802.11 standards

IEEE 802.11 is a set of media access control (MAC) and physical layer (PHY) specifications for wireless local area network (WLAN) in the 2.4 GHz, 5 GHz and 60 GHz frequency bands. The base version of the specification was released in 1997 followed by 802.11b and 802.11a in 1999. 802.11g was released in 2003 while 802.11n was drafted in 2009. The latest IEEE 802.11ac standard was released in 2014. The key features of 802.11 standards including the bandwidth, modulation and maximum data rates are tabulated in Table 2.1. I briefly introduce them one by one below.

Standard	Frequency band	Bandwidth size	Modulation	Maximum rate	Draft year
802.11	2.4 GHz	20 MHz	DSSS	2 Mbit/s	1997
802.11b	2.4 GHz	20 MHz	DSSS	11 Mbit/s	1999
802.11a	5 GHz	20 MHz	OFDM	54 Mbit/s	1999
802.11g	2.4 GHz	20 MHz	DSSS, OFDM	54 Mbit/s	2003
802.11n	2.4 GHz, 5 GHz	20 MHz, 40 MHz	OFDM	450 Mbit/s	2009
802.11ac	5 GHz	20/40/80/160 MHz	OFDM	1300 Mbit/s	2014
802.11ad	60 GHz	2.16 GHz	OFDM	7000 Mbit/s	Not yet

Table 2.1: IEEE 802.11 Standards

2.1.1 802.11a/b/g

802.11b is a direct extension of the original 802.11 standard defined in 1997 operating on the 2.4 GHz band together with other devices including microwave ovens, Bluetooth and cordless telephones. Its maximum achievable raw data rate is 11 Mbit/s. The bandwidth is 20 MHz and direct-sequence spread spectrum (DSSS) modulation scheme is utilized. A total of 11 channels with 3 non-adjacent channels can be used.

802.11a was proposed with orthogonal frequency-division multiplexing (OFDM) modulation in the physical layer hosted on the 5 GHz band. Compared with the crowded 2.4 GHz band, the 5 GHz band is rarely used. 802.11a is able to provide a maximum raw data rate of 54 Mbit/s. However, with a higher carrier frequency band, 802.11a also incurs higher loss in the air. So the effective range of 802.11a coverage is smaller compared to 802.11b/g. Also the ability of 802.11a to penetrate walls is weaker with a smaller wavelength.

802.11g was introduced in 2003. It is on the same 2.4 GHz band as 802.11b, but with OFDM modulation as 802.11a. Its maximum raw data rate is also 54 Mbit/s.

2.1.2 802.11n

The IEEE 802.11n standard was initiated in January 2004 and finalized in 2008 with its formal publication in 2009. With the improved performance offered by 802.11n, the standard soon became widespread and popular. The 802.11n standard provides a much better performance and keeps pace with the rapidly growing speeds of wired network. 802.11n operates on both 2.4 GHz and 5 GHz bands. The maximum raw data rate can be achieved is 450 Mbit/s. In order to achieve this higher performance, quite a few new features have been incorporated into the 802.11n standard:

- **Use of MIMO:** MIMO is a technique that exploits multipath propagation opportunities (diversity) to improve wireless capacity. MIMO technology will be introduced in more detail shortly in Section 2.2. The data is split into a number of spatial streams and transmitted through different antennas to the antennas at the receiver. The current 802.11n standard allows up to four simultaneous spatial streams.

- **Optional 40 MHz bandwidth:** An optional mode for 802.11n is to utilize a double-sized channel bandwidth. Previous Wi-Fi systems use overlapped 20

MHz bandwidth for each channel. This new protocol has the option of using 40 MHz instead. The main trade-off for this is that there are fewer channels can be used simultaneously. At 2.4 GHz, three 20 MHz channels can be used at the same time without interference with each other. For a 40 MHz bandwidth, only one channel can be supported. Thus the choice of whether to use 20 or 40 MHz has to be made by the number of simultaneous transmitting/receiving devices. If there is only one pair of devices transmitting and receiving, 40 MHz can be used as otherwise the bandwidth is not fully utilized and wasted.

- **Diversity with multiple antennas:** For 802.11n, the number of antennas attached to one AP has been significantly increased and the idea of beamforming and diversity are realized with these multiple antennas. The increasing number of antennas means larger diversity can be achieved in the presence of multipath reflections.

2.1.3 802.11ac

802.11ac is the latest 802.11 specification drafted for Wireless Local Area Networks (WLANs). The main goal of the new 802.11ac amendment is to significantly increase the throughput. This higher rate requirement is motivated by the continuing trend to transition devices from wired cable links to wireless links and by the emergence of new applications with ever higher throughput requirements. Existing 802.11 technologies operate in both the 2.4 GHz and 5 GHz bands. 802.11ac operates strictly on the 5 GHz band, supporting backwards compatibility with other 802.11 technologies operating on the same band. 802.11ac is able to provide a raw data rate of 1.3 Gbit/s. To achieve its goals, 802.11ac relies on a number of improvements in both the physical and MAC Layers. The physical layer improvements include:

- **Increased bandwidth per channel:** 802.11ac supports channel bonding to achieve up to 160 MHz bandwidth, four times the bandwidth supported by 802.11n.

- **Increased number of spatial streams:** up to eight spatial streams are supported, further increasing the overall throughput.

- **Higher-order modulation - 256 Quadrature Amplitude Modulation (QAM):** four times denser than 802.11n, further increasing the bit rate density.

- **Multi-User Multiple Input and Multiple Output (MU-MIMO):** supports simultaneous transmissions to multiple clients. Up to four distinct clients can receive different data simultaneously from a single AP. This is a big step in Wi-Fi networking as in the first time, simultaneous transmissions of different streams to multiple clients become possible.

The MAC layer improvement includes larger maximum size of aggregate MAC Protocol Data Units (MPDUs). Also, the Request to Send/Clear to Send (RTS/CTS) scheme has been refined to allow more efficient dynamic bandwidth operation.

2.1.4 802.11ad and 802.11af

IEEE 802.11ad is a new physical layer defined for the millimeter wave 60 GHz band. This high frequency band has a very different propagation characteristics compared to 2.4 GHz and 5 GHz. The peak transmission rate can be up to 7 Gbit/s. However, the attenuation of 60 GHz signal in the air is very large and it gets absorbed very quickly in the wall. Directional transmission needs to be employed for medium/long range transmissions and only LOS transmission is possible.

802.11af, also referred as White-Fi is drafted in 2014 which allows WLAN operation in the TV white space bands between 54 and 790 MHz. It employs the cognitive radio technology to transmit and avoid interference to the primary users. The transmission range is large compared to 2.4 GHz and 5 GHz bands. The channel size is usually 6-8 MHz. MU-MIMO similar to 802.11ac can be employed to transmit up to four simultaneous steams. With four spatial streams and four 8 MHz channels bonded, 802.11af can support a maximum raw date rate of up to 500 Mbit/s.

2.2 MIMO

The earliest idea related to MIMO were proposed in1970 and later in 1993 the concept of spatial multiplexing (SM) using MIMO was proposed by Paulraj and Kailath [77]. In 1996, Raleigh and Foschini refined and included new approaches [29, 83] to MIMO, which considers configurations that multiple transmit antennas are co-located at one transmitter to improve the link throughput.

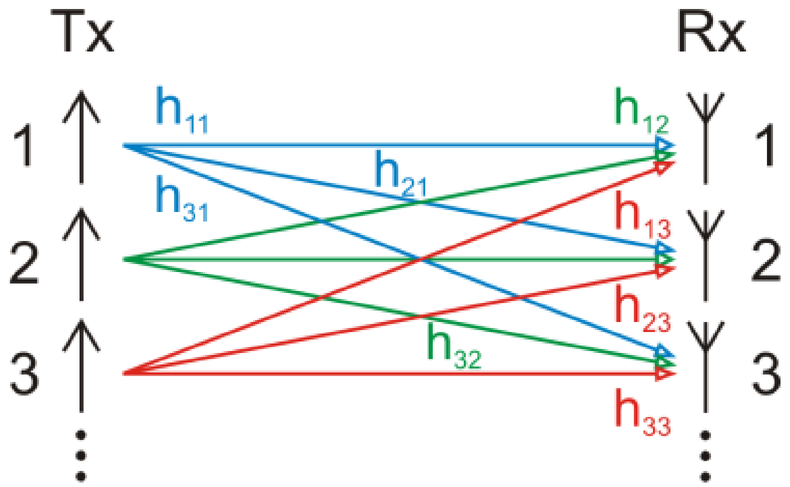


Figure 2.1: Basic MIMO concept: streams are transmitted and received by multiple antennas.

The first commercial system was developed in 2001 that used MIMO with Orthogonal frequency-division multiple access technology (MIMO-OFDMA) which supported both diversity coding and spatial multiplexing. In 2005, Airgo Networks developed a 802.11n precursor implementation based on the MIMO technology. Following that in 2006, Broadcom, Intel, and Marvell etc have fielded the MIMO-OFDM solution based on the pre-standard for 802.11n.

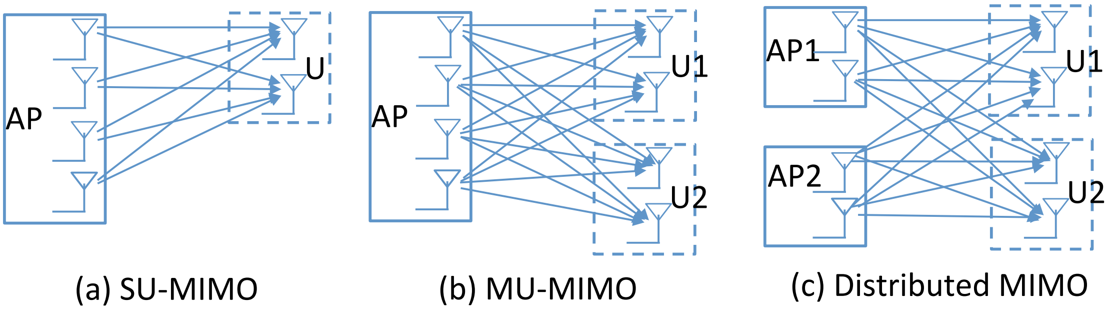
The basic idea of MIMO with channel model is illustrated in Figure 2.1. A transmitter sends multiple streams by multiple transmit antennas. The transmit streams go through a matrix channel which consists of all the paths between the N_t transmit antennas at the transmitter and N_r receive antennas at the receiver. Then, the receiver gets the received signal vectors by the multiple receive antennas and decodes the received signal vectors into the original information. A narrowband flat fading MIMO system is modeled as:

$$y = H * x + n \tag{2.1}$$

where y and x are the receive and transmit vectors, respectively; H and n are the channel matrix and the noise vector, respectively.

The function of MIMO can be further sub-divided into two main categories: beamforming and spatial multiplexing (SM).

- **Beamforming (precoding)** [106] is a signal processing that occurs at the transmitter. The signals are emitted from each of the transmit antennas with appro-



(a) SU-MIMO (b) MU-MIMO (c) Distributed MIMO

Figure 2.2: Different types of MIMO.

priate phase and amplitude weighting such that the signal to noise ratio (SNR) is maximized at the receiver side. The benefits of beamforming are to increase the received signal strength, by making signals transmitted from different antennas add up constructively. Note that precoding requires knowledge of *channel state information* (CSI) describing the channel between transmitter and receivers.

- **Spatial multiplexing** [34] requires a multiple-antenna configuration. In spatial multiplexing, several streams are transmitted from different transmit antennas in the same frequency channel. If these signals arrive at the receiver antenna array with sufficiently large spatial distances, the receiver can separate these streams. Spatial multiplexing is a very powerful technique for increasing channel capacity at higher SNR. The maximum number of spatial streams is limited by the number of antennas at the transmitter and receiver.

MIMO technology has evolved from Single-User MIMO (SU-MIMO) to Multi-User MIMO (MU-MIMO) and distributed MIMO in the latest few years. The architecture differences are illustrated in Figure 2.2. Traditional MIMO is actually SU-MIMO which only supports a single user reception (the user may have multiple antennas). Both MU-MIMO and distributed MIMO support simultaneous transmission to multiple clients (users) at the same time. The details of MU-MIMO and distributed MIMO are described below.

2.2.1 Multi-User MIMO

In a MU-MIMO [30] system, an AP communicates with multiple users simultaneously. MU-MIMO downlink capacity can scale with the minimum of the number of AP antennas and the sum of antennas at the user side. This means that MU-MIMO can achieve

MIMO capacity gains with a multiple antenna AP and multiple single antenna users! This is of particular interest since in reality, the number of antennas is limited to 1-2 on small handheld devices. MU-MIMO has been employed in LTE and also the latest 802.11ac standard.

2.2.2 Distributed MIMO & Distributed Antenna System (DAS)

Distributed MIMO (MegaMIMO [81], AirSync [8]) is a set of advanced MIMO technologies where the available antennas are spread over a multitude of independent access points - each having one or multiple antennas. Here transmitting antennas are located at different APs, which run a fine-grained synchronization protocol for phase-coherent transmissions. In theory, such a system enjoys all the significant performance gains of a traditional MIMO system, and it may be deployed in an enterprise Wi-Fi like setup. Phase and time synchronization at all the APs and also sharing their data on the wired backhaul network are the basic requirements to enable distributed MU-MIMO.

The distributed antennas of a DAS (distributed antenna system) system merely radiate the received RF signal (from the AP) within the AP's contention domain. Wi-Fi [22, 127] and mobile cellular networks [4, 118] have long used conventional DASs to provide improved wireless coverage indoors, especially in larger venues, such as convention centers or stadiums. A conventional DAS simply broadcasts the *same* signal from all antennas, with some designs even using "leaky feeder" coaxial cable that radiates RF energy continuously along the length of the cable [88]. Thus conventional DASs preclude the use of MU-MIMO and its associated spatial multiplexing gains. In a DAS, the antennas of an AP are distributed (extended) spatially with the help of RF or optical cables. Each of the distributed antennas may house one or more antennas and is controlled by the same AP with no change to the AP's hardware. An advantage of the DAS system for localization is that all the antennas are fully time synchronized so TDoA can be directly applied without any time synchronization issue. On the other hand, the APs in Distributed MIMO system are usually time synchronized with 5-20 ns synchronization error.

2.3 Indoor localization

For localization in an outdoor environment, GPS works extremely well. Differential GPS (DGPS) [71] now allows civilian GPS units to obtain an accuracy of 10 meters or better. Unfortunately, the signal from the GPS satellites is too weak to penetrate through the walls of most buildings, making GPS useless for indoor localization. In the last two decades, many different technologies have been employed to provide indoor localization services. Related work on indoor localization broadly groups into the following categories:

2.3.1 Ultrasonic and infrared based approaches

A lot of early work in indoor localization have employed infrared and ultrasonic infrastructures.

The Active Badge [115] system is a pioneer work leveraging infrared transmissions from badges carried by users for localization. The small infrared device worn by a user transmits a globally unique infrared signal for localization every 10 s or on demand. These transmissions are collected by the sensors placed at fixed locations in the building and sent to the central server for processing. The performance of infrared based location system is degraded in locations with fluorescent lighting or sunlight. Also the transmission range of infrared transmission is only several meters, limiting its application to small-sized rooms. A dense deployment of sensors are required to cover the whole area. The location accuracy achieved is on an order of room-size.

The Bat [37, 116] system employs ultrasound signals for better accuracy compared to Active Badge system because of the much lower sound transmission speed in the air compared to infrared signals. A short pulse of ultrasound is sent from the transmitter attached to the objects or carried out by human. The receivers are mounted at known locations on the ceiling. The transmitters and receivers are synchronized by a central controller. The synchronization request is sent to the transmitter via a short-range radio and sent to the ceiling sensors through a wired network. With a minimum of two range readings, the location estimate can be obtained. With more readings, even the 3D location information can be obtained. By finding the relative positions of multiple Bat transmitters attached to a single object, the orientation of the object can also be found. 720 receivers were deployed to cover an area of 1000 m^2 on three floors. The system is able to determine 75 objects' positions in one second with an average accuracy level of 3 cm in three dimensions. The accuracy level of Bat is quite amazing, but performance is very sensitive to the placement of sensors. A lot of dedicated devices need to be installed for Bat to work, making scalability and cost disadvantages for the Bat system.

Cricket [79] employs a combination of ultrasound and radio for accurate indoor localization and so is complementary to the Bat system. The radio signal is used for time synchronization between the receivers. In contrast with Bat, there is no time synchronization needed between the transmitter and receivers, so time difference of arrival is employed for its location estimate. Cricket is able to achieve below 10 cm accuracy. However, same as the Bat system, it needs dedicated infrastructure installation and the range of ultrasound signal is limited to around 10 meters which is much smaller than the Wi-Fi coverage (50-100 meters in a typical office environment), so a quite dense deployment is needed. The new version of Cricket (v2) is programmed via TinyOS and it can be powered by both batteries and external power connector.

Guoguo [64] deploys an acoustic ranging infrastructure, where the innovation is at the APs that are designed to send out acoustic beacons for ranging. The acoustic beacons are imperceptible to humans. It employs the combined matrices of RMS delay spread to identify the NLOS channel condition and assign lower weight for the measurement from NLOS channel during localization processing. Guoguo achieves a 6-25 cm median location accuracy which is not a very surprising result for acoustic based location because of the low transmission speed (340 m/s in the air). A big problem for acoustic based localization is that the sound noise is quite common in a lot of environment and it greatly affects the localization accuracy. Small coverage is another concern for an acoustic based location system since walls block sound easily.

2.3.2 Camera and visible light based approaches

Epsilon [38, 56] employs visible light from smart LEDs coupled with custom light sensor receivers for localization. In order to avoid flicker to human eyes, the frequency needs to be chosen higher than 200 Hz and also stay away from the 50/60 Hz (sound frequency) interference zone. On the other hand, at the receiver side, the light sensors of the commodity phones we are using have very limited sampling frequency which is up to several hundreds hertz. Epsilon is able to achieve sub-meter accuracy. However, its usage is limited to LOS scenario and a device needs to have at least three LOS LED anchors. Furthermore, the device needs to be exposed to the light in order to be localized so a mobile phone in the pocket or bag will pose a problem for the system. Even when a person is holding the phone in hand, the body may block the light and the body reflection also adds noise to the localization result.

Luxapose [52] uses off-the-shelf cameras as receivers coupled with image processing techniques for indoor localization. The basic idea is to include several anchors with known position and the target in the same image. The anchors are LED luminaries with different frequencies assigned. Then based on the relative positions of the anchors and the target in the image, multiple distance constraint equations can be generated. With enough number of anchors in the image, Luxapose is able to solve all the constraint equations and finds out the location of the target. Luxapose is able to provide sub-meter level location accuracy and also the orientation information of the object which is not available in a lot of location systems. By comparing the coordinate axis in the image with the ground truth, Luxapose is able to determine the mobile's orientation to an accuracy of 3°. However, for Luxapose to localize a mobile, the mobile needs to record a image with at least three visible LED anchor points. It also needs good light conditions and takes around 10 s for one location estimate. Another problem for camera based localization is the severe privacy issue, and so it may not be a good option in public areas.

Because of the popularity and ubiquitousness of Wi-Fi, it is still preferred to host the localization system on the Wi-Fi infrastructure. The Wi-Fi based localization schemes can be summarized as three categories: received signal strength indicator (RSSI) [6, 13, 36, 48, 54, 66, 86, 99] and CSI [46, 96, 122, 125], angle of arrival (AoA) [23, 47, 50, 51, 55, 73, 74, 95, 98, 117, 120, 128] and time of arrival (ToA)/time difference of arrival (TDoA) [21, 24, 32, 44, 49, 58, 61, 67, 79, 105, 135].

2.3.3 RSSI and CSI signature based approaches

The most widely used physical layer information for indoor localization is the received signal strength indicator. While readily available from commodity hardware, RSSI readings are generally coarse and unstable due to multipath reflection, refraction and scattering in indoor environment. It is also affected by the temperature [9] and humidity factors [16] harming the accuracy of location service provided.

RADAR [6, 7] is a pioneer work builds "maps" of signal strength to multiple access points, achieving an accuracy in the order of meters. Later systems such as Horus [134] use probabilistic techniques to improve localization accuracy to around one meter when an average of six access points are within range of every location in the wireless LAN coverage area, but require large amounts of calibration. While some work has attempted to reduce the calibration overhead [36], mapping generally requires significant calibration effort. Other map-based work has proposed using overheard GSM signals from nearby towers [109], or dense deployments of desktop clients [5]. Recently, Zee [82] has proposed using crowd-sourced measurements to remove the calibration step, resulting in an end-to-end median localization error of three meters when Zee's crowd-sourced data is fed into Horus.

The second line of work using RSSI are techniques based on mathematical models. Some of these proposals use RF propagation models [84] to predict distance away from an access point based on signal strength readings. By triangulating and extrapolating using signal strength models, TIX [35] achieves an accuracy of 5.4 meters indoors. Lim *et al.* [62] use a singular value decomposition method combined with RF propagation models to create a signal strength map. They achieve a localization error of about three meters indoors. EZ [19] is a system that uses sporadic GPS fixes on mobiles to bootstrap the localization of many clients indoors. EZ solves the constraint equations using a genetic algorithm, resulting in a median localization error of between 2–7 meters indoors, without the need for calibration. Other model-based proposals augment RF propagation models with Bayesian probabilistic models to capture the relationships between different nodes in the network [66], and develop conditions for a set of nodes to be localizable [131]. Some other model-based proposals are targeted towards ad hoc mesh networks [12, 90, 78]. I discuss some of the most important work in detail below.

RADAR [6, 7] is a pioneering work from Microsoft Research that utilizes RSSI readings from multiple 802.11 APs to provide a localization service. The process of RSSI-based localization such as RADAR is divided into two parts: the offline training (survey) stage in which the RSSI signatures are collected and the online matching stage to estimate the target's location. RADAR measures the RSSI values deterministically at each location from multiple APs. On each radio map, the particular RSSI values are distributed as contour lines in the 2-D floorplan. The RADAR system requires that a client must collect data from three APs to obtain the location information. The more APs the client can overhear, the more accurate location service it can provide. However, moving furniture and/or large groups of people moving around necessitate a reconstruction of the RSSI database. The median accuracy of the RADAR system is around 2-3 meters.

Horus [134] is another localization system based on RSSI readings. Unlike RADAR to measure RSSI readings deterministically at each location, Horus stores the information about the RSSI distributions from the APs and adopts a probabilistic technique to estimate the user location. Horus employs location-clustering schemes to reduce the computational requirements of the algorithm. Horus also models the RSSI distributions received from APs to reduce the effect of temporal variations. Horus claims to achieve below 1 m median accuracy.

Nibble [13] proposed by Castro et al. is system built by UCLA to provide indoor location services. It uses signal to noise ratio (SNR) rather than RSSI as a signature. The commonly applied techniques inferring location are multilateration, nearest neighbor and Bayesian network. In multilateration, an object can infer its location by calculating its range from beacons with known locations. This works fine in outdoor environments, but the walls and obstacles in indoor environment makes this method challenging. The nearest neighbor method is very coarse. The Bayesian network method can incorporate more information such as location of the room and wall structures into the probability model. Nibble applies a Bayesian network to obtain the location of a client. Nibble is one of the first systems to use a probabilistic approach for indoor location estimation.

A slight departure from conventional approaches, Modellet [57] makes the case for a hybrid model combining reading-based and model-driven localization approaches to handle data diversity and density in large scale deployments. The key intuition is that with different signature densities and environments, different approaches should be chosen to yield best results. It is very common to have different signature densities with war-walking as pathway areas usually have much denser data than the inner parts of shops. The internal layouts of shops in a shopping mall may also differ significantly. The key observation in Modellet is that when the signature density is high, a reading-based method is more accurate, while model-driven scheme works better when very few data are available. The unifying localization framework proposed is able to address the challenges and achieve performance better than a single reading-based or model-driven based location approach alone.

In addition to the coarse RSSI information, later work has leveraged finer channel state information (CSI) for localization purpose. The channel response information on each subcarrier includes both amplitude and phase which can be utilized to provide a lot more information than RSSI readings. This CSI information is utilized to localize client and also prevent attacks.

PinLoc [96] employs CSI information for each OFDM subcarriers as a promising location signature. The CSI information is also available from the off-the-shelf Intel 802.11n 5300 card. PinLoc leverages the observation that multipaths exhibit stable patterns on the subcarrier CSI information when they combine at a given location, and these patterns can lend themselves to meter-scale localization. PinLoc is able to localize users to the correct 1m x 1m spot with a 89% mean accuracy, while incurring only 6% false positives, outperforming Horus, the most accurate RSSI based localization scheme.

CSITE [46] uses CSI magnitude measurements averaged in time over multiple frames to form a unique client signature for security purposes. CSITE is able to achieve a 90% detection rate with a 5% false positive rate. However, since RSSI is simply CSI magnitude information averaged across different subcarriers, per-subcarrier CSI approaches such as CSITE can be subverted by attackers with phased array software-defined radios capable of transmitting with independent beamforming weights on different subcarriers.

SecureArray [122, 125] employs the CSI information from the antenna array to generate AoA signatures of the incoming signals. This AoA signature is much more stable than the RSSI readings. With a novel random phase perturbation scheme applied, SecureArray is able to detect 100% of attacks while triggering a false alarm of 0.6% on the legitimate traffic. On the other hand, AoA signature also provides information directly related to the location of the client: the bearing of the client. With 6-8 antennas, the angle accuracy of a linear array is around $3-10^{\circ}$.

Many other works [13, 36, 48, 54, 66] either try to improve the accuracy or reduce the site survey efforts. However, in general, the RSSI and SNR readings from APs are not stable and are relatively coarse for localization purposes.

2.3.4 Combination of RSSI/CSI and sensor readings

Liu *et al.* [63] combine Wi-Fi RSSI fingerprinting with acoustic ranging information between smartphone users to increase accuracy. The key idea is to obtain accurate acoustic ranging estimates among peer phones, and then map this acoustic distance relationship jointly against the Wi-Fi signature map to increase the localization accuracy. So the Wi-Fi RSSI is used to obtain the rough absolute location estimate and the acoustic ranging estimate is used to apply the accurate distance constraint (relative location information) to increase the accuracy. With the help of these acoustic ranging constraints, the accuracy is improved from 6 - 8 m to around 1 m. While peer assistance is adopted for localization, more power and bandwidth will be consumed at the peers. Also in order to help the target, the locations of the peers are exposed which is a compromise of the privacy.

SAIL [68] leverages fine-grained CSI information coupled with inertial dead reckoning on smartphones to enable localization with a single AP. Dead reckoning is the process of calculating one's current location by using a previously determined location together with the help of the sensors in the mobile. The basic idea is to employ the dead reckoning method to obtain the distance travelled by the mobile and the ToF (time of flight) method to obtain the distances between the AP and the mobile. The three distances obtained yield a unique triangle. Together with the compass heading of the mobile, location estimate can be obtained. However, finding the offset between the phone's and user's headings is not trivial. The mobile's orientation may also be quite random when it is in use. ToF accuracy is highly dependent on the sampling rate of the hardware which is usually constrained by the bandwidth. Also multipaths greatly affect the performance of ToF estimates. With a sampling rate of 88 MHz, SAIL's localization accuracy is around 2-3 meters. The performance will be worse with a usual 40 MHz sampling rate and an environment with strong multipath propagation.

Zee [82] leverages crowdsourcing to eliminate the calibration stage for RSSI based localization. Zee employs the inertial sensors such as accelerometer, compass and gyroscope in the mobile to track them while simultaneously performing Wi-Fi scans. The only site input is a floorplan map with the pathway and barriers (e.g.,walls). Zee is able to track users without the users' initial location, stride length and phone placement information. The intuition of Zee is to combine the sensor information with the constraints imposed by the map, thereby filtering out infeasible location estimates over time and converging gradually to the true location. To speed up the convergence of particle filtering, Zee employs two techniques: motion estimation to estimate the step count and the approximate orientation of a device relative to the direction of walk, and Wi-Fi based initialization to make an initial guess of the device's starting point. Uncertainty in location tends to reduce with time as a user takes more turns which are physical constraints added. Zee also uses backward belief propagation scheme to reduce uncertainty in location at earlier times, post facto. When a location estimate is available, the corresponding Wi-Fi measurement is annotated with the estimated location, thereby adding a record to the Wi-Fi RSSI training data. With this scheme, Zee is able to build the training data set without any explicit effort from the user and the service provider. With the data set built with Zee, those RSSI-based location schemes are able to achieve around 3-5 m median accuracy which are comparable to the results achieved with a training data set that is explicitly measured.

LiFS [132] is another system similar to Zee trying to eliminate the training process that RSSI fingerprinting scheme requires. Site survey training is time-consuming, labor-intensive and vulnerable to environment dynamics. LiFS exploits user motions from the mobile devices to remove the site survey effort of the traditional RSSI based localization. The key intuition is that human motion can be applied to connect previously independent radio fingerprints. LiFS proposes a stress-free floorplan concept which finds out the walking distance between two locations. LiFS also constructs the fingerprint space by computing all-pair shortest paths of fingerprints which takes $O(n^3)$ running time while n is the number of fingerprints. The fingerprint space is then mapped on the stress-free floor plan with simple shift and linear transformation. LiFS is able to reduce the survey efforts significantly while still achieving reasonable accuracy level of 5-6 meters which is slightly worse than the traditional RSSI based localization method with RSSI fingerprint data measured. However, a big disadvantage for LiFS is the high computational load incurred. When the survey area is large or the signature density is high with a big n, it may take a long time for LiFS to generate the fingerprint space.

Unloc [110] is an unsupervised indoor localization system that bypasses the need of war-driving to collect RSSI signatures. It combines dead-reckoning, urban sensing and Wi-Fi-based localization into a single system. The key intuition is to detect the unique landmark signature with the sensor data collected from the accelerometer, compass and gyroscope within a Wi-Fi sub-space. With the Wi-Fi RSSI readings, all the processing can be restricted to a relatively small area, so the computational load is much reduced. Dead-reckoning is employed to roughly estimate the location of the landmark starting from a known reference. With the landmark location estimates from multiple users, the accuracy of the landmark locations can be improved and used to correct the dead-reckoning errors. The errors get reduced as more landmarks are discovered. It does not even require the floorplan and simultaneously computes the locations of users and landmarks in a manner that converges quickly. Unloc is able to achieve a less than 2 m median error without any pre-deployment.

SpinLoc [94] leverages the human body's attenuation of Wi-Fi signals when users spin around to estimate bearings to the APs. SpinLoc's angle estimate scheme is coarse with an error of up to 30°. How the mobile is held in the hand also greatly affects the performance of SpinLoc. On the other hand, asking a user to spin 360° for a location estimate is not always feasible. SpinLoc is able to achieve around 6.5 m accuracy with four audible APs.

Centaur [72] fuses RF and acoustic based ranging using a Bayesian inference framework to enable higher accuracy indoor localization. The basic idea is to use accurate acoustic constraint to improve the accuracy of RSSI based Wi-Fi localization. Centaur assumes a lot of modern devices such as laptops and mobiles have both Wi-Fi interfaces and speakers/microphones. It proposes a scheme named EchoBeep to improve the acoustic ranging scheme for the NLOS scenario. Centaur is able to achieve around 1 m accuracy compared to 3-4 m accuracy with Wi-Fi alone. The processing time is relatively long for Centaur ranging from seconds to minutes due to high computational load.

2.3.5 AoA based approaches

Because of the limited number of antennas on a single AP, AoA scheme was previously not popular for Wi-Fi indoor localization. With the new 802.11n and 802.11ac standards, multi-antenna APs become common in enterprises, which enables an AoA approach to localization. Wong *et al.* [117] investigate the use of AoA and channel impulse response (CIR) measurements for localization. While they have demonstrated positive results at a very high SNR (60 dB), typical wireless LANs operate at significantly lower SNRs, and the authors stop short of describing a complete system design of how the ideas would integrate with a functioning wireless LAN. Niculescu *et al.* [73] simulate AoA-based localization in an ad hoc mesh network. AoA has also been proposed in CDMA mobile cellular systems [128], in particular as a hybrid approach between TDoA and AoA [23, 120], and also in concert with interference cancellation and ToA [104]. Much other work in AoA use the technology to solve similar but materially different problems. Geo-fencing [98] utilizes directional antennas and a frame coding approach to control APs' indoor coverage boundary. Patwari *et al.* [76] propose a system that uses the channel impulse response and channel estimates of probe tones to detect when a device has moved, but do not address location. Faria and Cheriton [27] and others [10, 65] have proposed using AoA for location-based security and behavioral fingerprinting in wireless networks. Chen *et al.* [17] investigate *post hoc* calibration for commercial off-the-shelf antenna arrays to enable AoA determination, but do not investigate localization indoors. I present several representative AoA localization systems below.

Wong et al. [117] investigate the use of a CIR method with MIMO to provide an indoor localization service. They utilize a CIR method to identify the angle of arrival of the received signal that corresponds to the transmission path directly from transmitter to receiver. This is not easy as in the NLOS situation, the received power of the direct path arrival is very low. It is also challenging to resolve the multipath arrivals in time, as arrivals with similar propagation distance will overlap. These channel measurements characterize the CIR between each of the measurement system transmit and receive antennas. For a 4x4 MIMO system, there are 16 channels and therefore 16 CIRs, one for each transmitter receiver antenna pair. The earliest arrival component in the CIR is detected and used to determine the location AoA for that arrival. The intuition here is simple: the direct path is shorter than the reflection path. The maximum likelihood algorithm does this by simply selecting the earliest detectable component in the CIR and estimating its AoA. The SAGE algorithm [28] is a general parameter estimation algorithm that attempts to estimate the various multipath arrivals in the channel, obtaining channel parameters for each of the arrivals in terms of the complex amplitude, delay and AoA. The AoA of the mobile at the AP is selected as that corresponding to the arrival with the smallest delay. While this method is quite interesting, they require a long training sequence and high SNR value (60 dB in their experiment) which is not realistic in real life.

Lang et al. [55] propose to combine both AoA and RSSI methods to localize indoor clients. However, their method of obtaining AoA information in real indoor environment is not realizable: the AP scans the surrounding 360° in a period of several seconds to detect transmissions. Wireless channel is shared by all the clients and each client takes turn to transmit and receive. The transmission of one packet from one client is in the time scale of milliseconds and when the transmission starts is not predictable. So the transmission may not be able to be captured by this kind of slow scanning method. What makes it worse: the AoA obtained may not be the direct path bearing and may be one of the multipath reflections AoAs.

Phaser [31] extends ArrayTrack [123, 124] to work on commodity hardware. By sharing one antenna between two Intel 5300 802.11n cards [45] each with three antennas, Phaser successfully synchronizes the two radio cards to form a 5-antenna array. Phaser is a general purpose platform for any phase-based signal processing hosted on commodity hardware. With four 5-antenna APs operating on 5 GHz band, Phaser is able to achieve 1-2 meter median accuracy in a crowded student office environment.

PinPoint [47] proposes AoA processing with cyclo-stationary analysis to identify the direct path. PinPoint employs the known repeating patterns within wireless signals to form unique signature for every signal type. This unique repeating patterns such as long training symbol in the preamble is used to self-correlate to find the first peak identified as the direct path. However, when the direct path and reflection path are close to each other (less than one sample distance), it is not possible for this correlation scheme to differentiate the direct path and the reflection path. And in a typical indoor environment, the path difference between the direct path and reflection path is usually small. So the correlation scheme proposed in PinPoint does not work in reality. All the figures presented in the paper have the direct path and reelection path separated at least by 70 m (corresponding to 0.23 ms) which is very unlikely to happen in indoor environment. In order to minimize the uncertainty of LOS path identification with the scheme proposed, PinPoint needs to collect multiple packets which may impede the realtime response of the system. PinPoint achieves a median accuracy of 0.97 m.

LTEye [51] localizes LTE clients from their uplink signal transmissions using synthetic aperture radar (SAR). Mechanically rotating antennas are used to form the circular SAR. LTEye also propose a scheme to identify the shortest path (not necessarily the direct path) by calculating the time delay of the signal at one particular direction with phase information obtained at each subcarrier. LTEye assumes a separation of the direct path and reflection path and then calculate the ToA to determine the shortest path. However, it is much more difficult to separate the direct path and reflection path signals than obtaining the ToA information of the mixed signals. LTEye is able to achieve a median bearing error of $7-10^{\circ}$.

Ubicarse [50] takes one step further from LTEye by making the smartphone emulate a SAR through user induced motion. To emulate a SAR, the mobile needs to rotate exactly following a circular trajectory. However, it is difficult for a human to rotate the mobile devices strictly in a circle. Unicarse leverages the opportunity that there may be two antennas in a mobile and the relative distance between the two antennas is fixed. So the relative channel between these two antennas is employed to emulate a SAR to solve the non-circular movement issue. However, another big problem Ubicarse does not address is that the rotation speed of the mobile must be constant in order to emulate a SAR. Ubicarse is able to emulate a perfect circle but the varying rotation speeds may still degrade the performance greatly. Ubicarse is able to achieve a median location error of 39 cm in experiments where the rotation speed is apparently constant. The performance may be much worse if the rotation speeds vary during one round of rotation, however.

CUPID [95] combines AoA estimation with the user's mobility pattern to identify the direct path angle arrival for localization. The key idea follows ArrayTrack by identifying the more stable AoA peak as the direct path peak. Because of the limited number of antennas used in CUPID, it is able to achieve a 20° angle error. Because of this low AoA resolution and high angle error, it may also experience the scenario when both direct path and reflection paths are not stable. In this scenario, CUPID employs dead-reckoning and calculates the angle change to identify the direct path. The idea is that with a large distance of movement, CUPID can calculate the angle change from the sensor readings and match this value with the angle change on the AoA pseudospectrum. The accuracy will be improved with more antennas attached to the AP.

2.3.6 ToA/TDoA based approaches

ToA) and TDoA schemes are very popular with Ultra-wideband (UWB) systems [44, 67, 49, 135, 21, 61]. The resolution of the ToA/TDoA based localization is restricted by the bandwidth. While UWB has a bandwidth larger than 500 MHz, the ToA/TDoA resolution is very high while Wi-Fi does not provide a more than 20 MHz bandwidth until 802.11n. With such a small bandwidth, the direct path and reflection path signals are usually too close to be separated making it difficult to provide accurate ranging information. ToA and TDoA can be converted to each other without ambiguity and are proven to be equivalent [25]. The basic idea is inherited from military outdoor RADAR system. The basic concept of outdoor RADAR system is that the RADAR device has a transmitter that sends a pulse to the object to measure the distance. The pulse bounces off of that object and the receiver, physically located within the RADAR device, receives the pulse reflection. The RADAR device measures the time difference from the time it sent the pulse to the time it received the pulse. This time difference is related linearly to the distance to the object. Thus, the distance can be easily determined. Ultra-wideband radios, which have been commercialized [105], can discern sub-nanosecond time-of-flight, but performance decreases in the presence of multipath propagation and direct path blockage [24], and practical UWB radios use very low data rates and power due to government regulator rules, severely limiting performance.

Zhou *et al.* [135] propose a UWB localization system without time synchronization between the target and the receivers. If the target and receivers are synchronized, the absolute range between them can be found easily. However, it is not possible to achieve tight time synchronization between the target and the receivers in most of the applications. On the other hand, it is less challenging to have all the receivers synchronized. In this case, the relative ranges of the receivers with respect to the target can be found and the problem is converted into a TDoA problem. In this work, a scheme similar to [26] is proposed to remove time synchronization between the target and receivers and between the receivers. The key idea is to have another transmitter with known locations serving as a reference and trigger the transmission with this reference. Whenever the transmission from this transmitter is received by the receivers and the target, the target will start another transmission and all the receivers will receive the second transmission from the target. By measuring the time difference between the receptions of the two signals at the receivers, the distance of the target and the receivers can be obtained. The achievable median accuracy is around 10 cm. There are still several problems with this scheme. The time gap between the reception and transmission at the target side needs to be measured very accurately and this value is highly dependent on the device itself. So one calibration is only working for one target device. Also the transmitter chosen needs to have direct LOS paths to all the receivers and the target which is not always possible.

Traditionally, MUSIC and ESPRIT algorithms are applied to obtain AoA information. Li and Pahlavan [59] propose using MUSIC in the frequency domain for ToA estimation. Because of popularity of OFDM and MIMO technologies in Wi-Fi, subcarrier CSI is available which makes super-resolution processing of ToA possible.

In [32], Golden *et al.* utilize the ToA method to obtain indoor location for client. A similar reference observer as in [26, 135] is chosen to handle the time synchronization problem between the transmitter and receivers. Because of the strong multipaths in indoor environment, ESPRIT [87] algorithm is employed to decompose the direct path and multipaths. The achievable median range accuracy is between 1-6 m with a 40 MHz sampling rate.

Synchronicity [126] uses MUSIC super-resolution techniques to differentiate the direct path and reflection paths. TDoA is employed in Synchronicity to bypass the time synchronization between the target and the receivers. DAS is employed to time synchronize all the receivers. One contribution of Synchronicity is the spectrum identification scheme proposed to retrieve useful information from an inaccurate MUSIC ToA profile. Synchronicity is able to achieve 2-3 m accuracy while its performance is limited by the radio bandwidth.

A recent independent work appearing in the literature, Splicer [119] employs a similar idea as ToneTrack of combining CSI from multiple channels. While Splicer combines CSI for a more accurate power delay profile, ToneTrack uses the combined CSI to further increase the resolution of super-resolution MUSIC for more accurate localization, integrating the CSI combination process with super-resolution ToA estimation.

JADE [108, 107] jointly estimates angle and delay of all the arriving multipath signals. Compared to MUSIC, JADE is able to present us information in both time and space at the same time but the computational load is much higher.

2.3.7 GPS-assisted approaches

EZ [19] employs genetic algorithms to triangulate users between Wi-Fi APs coupling with sporadic GPS fixes. The key advantage of EZ is that it does not require any knowledge of the environment including the location of the AP and the floorplan. EZ does not require any explicit user participation either. With the RSSI information recorded at multiple APs when the mobile moves to enough number of different locations, a localizable structure (the entire set of locations can be translated, rotated and reflected but not distorted in any manner if all distances are to be preserved) can be found. Then with three true, no-collinear locations obtained opportunistically, when GPS enabled mobile devices occasionally obtain GPS readings at the edges of the indoor environment such as entrances and windows, EZ is able to determine the client's absolute location. The computational load of EZ is large. On the other hand, EZ is able to achieve 2-7 m median localization error which is slightly worse compared to traditional RF fingerprinting based localization schemes.

COIN-GPS [75] explores the possibility of using GPS-based direct localization in some indoor environments. It is well known that GPS receivers do not work indoors because of the extremely weak signals caused by building shells and also the strong multipaths. COIN-GPS is a piece of work trying to break this limit. It employs a high-gain directional antenna at the front-end of the GPS receiver and computes the location estimates in the cloud. By steering the directional antenna towards different directions, it is possible to have a higher SNR in a certain direction and also the multipath effects are reduced. The indoor GPS signals are too weak to be decoded for timestamps and ephemeris. With the cloud-offload scheme, COIN-GPS eliminates the need of decoding the GPS signal and gets the ephemeris data through a web service. The signal processing in COIN-GPS is costly in terms of both energy and computational load which makes cloud processing an appropriate choice. COIN-GPS is able to achieve a median localization error of 9.6 m indoors, where all normal GPS receivers fail. COIN-GPS may not work in reality because of the requirement of high-gain directional antenna and the coarse accuracy performance. However, it is well appreciated that GOIN-GPS tries to overcome the constraints and make GPS localization work the first time in the indoor environment.

2.3.8 FM radio based approaches

Another line of work leverages FM radio for indoor localization owing to its lower frequency and hence better robustness to penetration, multipath and distance of transmission.

Chen *et al.* [18] leverage FM radio fingerprinting for localization purposes. Because the frequency band (88 -108 MHz) is much lower compared to Wi-Fi (2.4 GHz and 5 GHz), FM signals are less susceptible to human presence, multipath and fading, they exhibit exceptional wall penetration capability and very less variation over time. One important observation from this paper is that the localization error of FM and Wi-Fi signals are independent. Because of this, FM and Wi-Fi schemes can be combined for localization which increases the accuracy by 83% compared to when only Wi-Fi signals are used as signatures for localization. However, as a signature based location system, the performance is greatly affected by environmental changes such as the presence of people and movement of furniture. Training and updating the signature database takes time and effort. Another disadvantage is that not many devices are now equipped with FM radios.

ACMI [133] is another FM-based localization system which does not require site survey (training) to store the RSSI signatures at each single spot in a building. The intuition is that FM signals are much more stable compared to Wi-Fi signals so the data base can be built purely based on the signal propagation model. ACMI proposes a practical model to predict the RSSI distribution only using the publicly available FM transmissions and the floorplan of a building. This approach saves expenditure in both time and cost for site survey. However, the accuracy level of the system is limited to around 6 m and 89% room identification using eight FM broadcast signals.

2.3.9 RFID based approaches

PinIt [112] leverages antenna motion to emulate an antenna array for AoA information. This AoA information is employed as a unique signature similar to the RSSI reading. PinIt employs the intuition that when two RFID tags are closer to each other, their AoA profiles are more similar to each other. PinIt uses the dynamic time warping (DTW) method to identify a spatial similarity between two AoA multipath profiles of nearby RFID devices. By calculating the similarity of the the AoA profiles between the target and the reference, PinIt is able to localize the client with a median accuracy of 11 cm. Compared to the RSSI signature, AoA signature is more stable but is still vulnerable to the layout changes and furniture movements in the indoor environment.

RF-Compass [111] employs RFIDs located on a robot to localize a given object with RFID attached. It introduces a new RF localization algorithm formulated as a space partitioning optimization problem. With any two tags attached to the robot, RF-Compass is able to partition the space into two equal parts based on which tag is closer to the target. With multiple tags on the robot, the target is restricted to within a small region. RF-Compass leverages the consecutive moves of the robot to generate more space partitions, further refining the accuracy of its estimates. With multiple RFID tags attached to the object, RF-Compass is also able to find the object's orientation in addition to its location. RF-Compass is able to achieve a median accuracy of 2.76 cm in center position and 5.77° in orientation.

RF-IDraw [113] improves the accuracy of ArrayTrack by increasing the separation between antennas in an antenna array. By increasing the separation, the resolution of the AoA beam is increased at the cost of more ambiguity. RF-IDraw proposes a new antenna array placement strategy with only four antennas: two in the middle with half wavelength separation to remove the ambiguity and two far away from each other to increase the resolution. However, when there are a lot of multipaths, RF-IDraw's performance will be greatly degraded. RF-IDraw is able to track the trajectory shape of the users writing with a median accuracy of 3.7 cm and a character recognition success rate of 96.8%.

Tagoram [130] is another localization system based on RFID. It is observed that the COTS RFID product is able to provide fine-grained resolution (0.09°) in detecting the phase of received RF signals. Tagoram exploits the tag's mobility to build a virtual antenna array with readings from a few antennas over a time window. With the Differential Augmented Hologram (DAH) scheme, Tagoram relaxes the assumption of knowing the tag's track trajectories. Tagoram is able to achieve below one centimeter accuracy in the Lab with known linear and circular tracks while 2-10 cm accuracy in real luggage tracking deployments at two airports. This is the best accuracy achieved until now with an RFID based localization system that has knowledge of a tag's trajectory. Tagoram also studies the effect of multipath and thermal noise on localization performance.

Chapter 3

Angle-of-arrival based localization

The proliferation of mobile computing devices continues, with handheld smartphones, tablets, and laptops a part of our everyday lives. Outdoors, these devices largely enjoy a robust and relatively accurate location service from GPS satellite signals, but indoors where GPS signals do not reach, providing an accurate location service is quite challenging.

Furthermore, the demand for accurate location information is especially acute indoors. While the few meters of accuracy GPS provides outdoors are more than sufficient for street-level navigation, small differences in location have more importance to people and applications indoors: a few meters of error in estimated location can place someone in a different office within a building, for example. Location-aware smartphone applications on the horizon such as augmented reality, building navigation, social networking, and retail shopping demand both a high accuracy and a low response time. A solution that offers a centimeter-accurate location service indoors would enable these and other exciting applications in mobile and pervasive computing.

Using RF for location has many challenges. First, the many objects found indoors near APs and mobile clients reflect the energy of the wireless signal in a phenomenon called *multipath propagation*. This forces an unfortunate tradeoff that most existing RF location-based systems make: either model this hard-to-predict pattern of multipath fading, or leverage expensive hardware that can sample the wireless signal at a very high rate. Most existing RF systems choose the former, building maps of multipath signal strength [6, 7, 109, 134], or estimating coarse differences using RF propagation models [35, 62], achieving an average localization accuracy of between one meter [134] and meters: too coarse for a lot of indoor applications at hand.

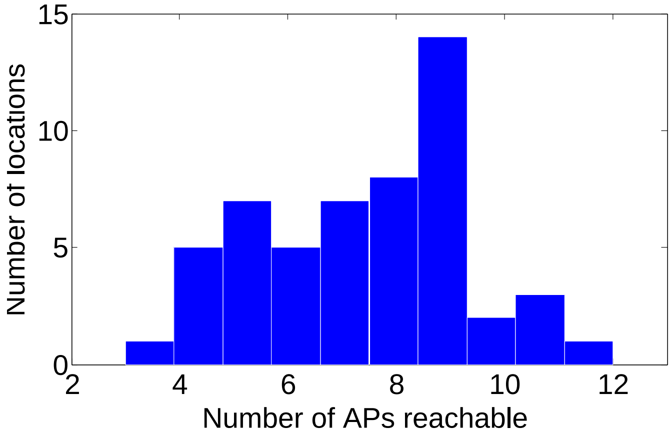


Figure 3.1: Number of production Wi-Fi access points (APs) reachable from every client location in our experimental testbed. Transmissions from most locations reach seven or more production APs.

Systems based on ultrasound sensors such as Bat [116] and Cricket [79] have achieved accuracy to the level of centimeters, but usually require dedicated infrastructure to be installed in every room in a building, an approach that is expensive, time consuming, and requires maintenance effort. So we still prefer the location system to be hosted on the existing Wi-Fi infrastructure.

Recently, however, two new opportunities have arisen in the design of indoor location systems:

1. Wi-Fi APs are incorporating ever-increasing numbers of antennas to bolster capacity and coverage with MIMO techniques. In fact, it is expected that in the future, the number of antennas at the AP will increase several-fold, to meet the demand for MIMO links and spatial multiplexing [3, 102], which increase overall capacity. Indeed, the latest 802.11ac standard specifies eight MIMO spatial streams, and 16-antenna APs have been available since 2010 [129].

2. Meanwhile, Wi-Fi AP density remains high: with my experimental infrastructure excluded, transmissions from most locations in my testbed (a typical student office in the university) reach seven or more production network APs as shown in Figure 3.1, with all but about five percent of locations reaching five or more such APs. Furthermore, by leveraging the signal processing that is possible at the physical layer, an AP can extract information from a single packet at a lower

SNR than what is required to receive and decode the packet. This allows even more APs to cooperate to localize clients than would be possible were the system to operate exclusively at higher layers.

This chapter describes *ArrayTrack*, an indoor localization system that exploits the increasing number of antennas at the APs to provide fine-grained location for mobile devices in an indoor setting. When a client transmits a frame, multiple multi-antenna ArrayTrack APs overhear the transmission, and each compute the AoA information from the clients' incoming frame. Then, the system aggregates the APs' AoA data at a central backend server to estimate the client's location. While the AoA technique is already in wide use in radar and acoustics, the challenge in realizing this technique indoors is the presence of strong multipath RF propagation in these environments. To address this problem, I introduce a novel scheme to eliminate the effects of multipath, even in the relatively uncommon situations when little or no energy arrives on the direct path between client and AP. ArrayTrack advances the state of the art in localization in three distinct ways:

1. Multipath reflection is a well-known challenge for AoA based indoor localization. To mitigate the effects of indoor multipath propagation, ArrayTrack contributes a novel *multipath suppression* algorithm to effectively remove the reflection paths between clients and APs. By employing the natural movements of human body, ArrayTrack is able to identify the more stable direct path for localization.

2. Upon detecting a frame, an ArrayTrack AP quickly switches between sets of antennas, synthesizing new AoA information from each. I term this technique *diversity synthesis*, and find that it is especially useful to increase the effective number of antennas in the case of low AP density.

3. ArrayTrack's system architecture centers around parallel processing in hardware, at APs, and in software, at the database backend, for fast location estimates. The packet detection is implemented in FPGA and only a tiny part of the preamble is transferred from the AP to the backend for AoA processing. The computational load is small at the backend and the whole process is completed within 0.1 s.

ArrayTrack is implemented on the Rice WARP FPGA platform [85] and evaluated in a 41-node network deployed over one floor of a busy office space. Experimental results in this setting show that using just three APs, ArrayTrack can localize clients to a median 57 cm and mean one meter accuracy. With six APs, ArrayTrack achieves a median 23 cm and mean 31 cm location accuracy, localizing 95% of clients to within 90 cm. At the same time, ArrayTrack is fast, requiring just 100 milliseconds to produce a location estimate. To my knowledge, these are the most accurate and responsive location estimates until 2013 for a Wi-Fi based location system that does not require infrastructure except a normal density of nearby Wi-Fi APs. Furthermore, ArrayTrack's performance is robust against different antenna heights, different antenna orientation, low SNR and collisions.

In the next section, I detail ArrayTrack's design. An implementation discussion (§3.2) and performance evaluation (§3.3) then follow.

3.1 Design

ArrayTrack's design is described as information flows through the system, from the physical antenna array, through the AP hardware, and on to the central ArrayTrack server, as summarized in Figure 3.2. First, ArrayTrack leverages techniques to detect packets at very low signal strength, so that many APs can overhear a single transmission (§3.1.1). Next, at each AP, ArrayTrack uses many antennas (§3.1.2) to generate an *AoA spectrum*: an estimate of likelihood versus bearing (§3.1.3), and cancels some of the effects of multipath propagation (§3.1.4). Finally, the system combines these estimates to estimate location (§3.1.5), further eliminating multipath's effects.

3.1.1 Packet detection and buffer management

To compute an AoA spectrum for a client, the AP only need to overhear a small number of frames (between one and three, for reasons that will become clear in Section 3.1.4) from that client. For ArrayTrack's purposes, the contents of the frame are immaterial, so ArrayTrack can process control frames such as acknowledgments, and even frames encrypted at the link layer.

The physical-layer preamble of an 801.11 frame contains known short and long training symbols, as shown in Figure 3.3. I use a modified version of Schmidl-Cox [91]

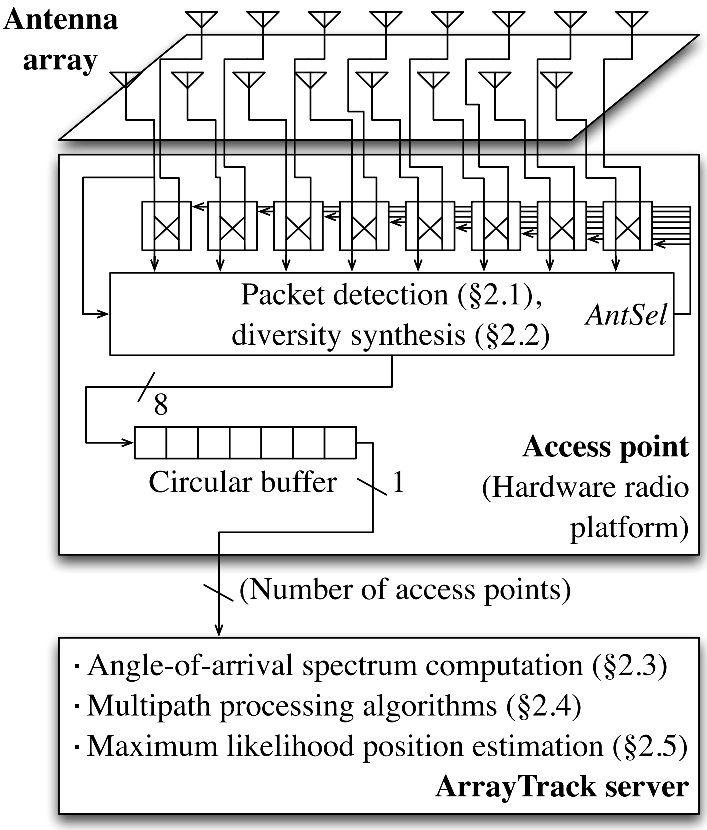

Antenna array
Packet detection (§2.1), diversity synthesis (§2.2)
AntSel
8
Circular buffer
1
Access point
(Hardware radio platform)
(Number of access points)
· Angle-of-arrival spectrum computation (§2.3)
· Multipath processing algorithms (§2.4)
· Maximum likelihood position estimation (§2.5)
ArrayTrack server

Figure 3.2: ArrayTrack's high-level design for eight radio front-ends, divided into functionality at each ArrayTrack access point and centralized server functionality. For clarity, the transmit path functionality of the access point is omitted.

detection to detect an incoming frame's short training symbols. Once the detection block senses a frame, it activates the diversity synthesis mechanism described in the next section and stores the samples of the incoming frame into a circular buffer, one logical buffer entry per frame detected.

Since it does not require even a partial packet decode, ArrayTrack can process any part of the packet, which is helpful in the event of collisions in a carrier-sense multiple access network (§3.3.3.5). ArrayTrack detects the preamble of the packet and records a small part of it. In principle, one time domain packet sample would provide enough information for the AoA spectrum computation described in Section 3.1.3. However, to average out the effects of noise, ArrayTrack employs 10 samples (I justify this choice in Section 3.3.3.3). Since a commodity Wi-Fi AP samples at 40 MHz/second, this implies that I need to process just $25 \times 10 = 250$ nanoseconds of a packet's samples, under 1.5% of an Wi-Fi preamble's 16 μs duration.

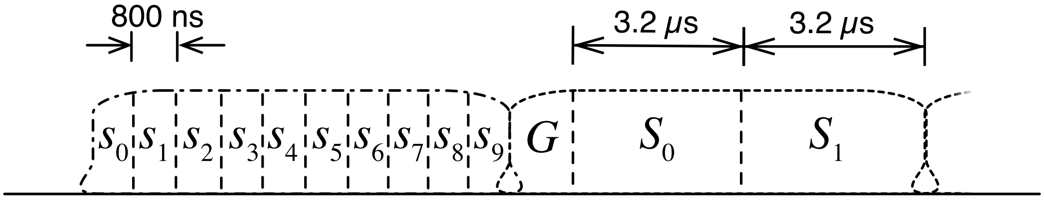


Figure 3.3: The 802.11 OFDM preamble consists of ten identical, repeated *short training symbols* (denoted $s_0 \ldots s_9$), followed by a *guard interval* (denoted G), ending with two identical, repeated *long training symbols* (denoted S_0 and S_1).

3.1.2 Diversity synthesis

Upon detecting a packet, most commodity APs switch between pairs of antennas selecting the antenna from each pair with the strongest signal, a technique called *diversity selection*. This well-known and widely implemented technique improves performance in the presence of destructive multipath fading at one of the antennas, and can be found in the newest 802.11n MIMO access points today. It also has the advantage of not increasing the number of radios required, thus saving cost at the AP.

ArrayTrack seamlessly incorporates diversity selection into its design, synthesizing independent AoA data from both antennas of the diversity pair. This technique is termed *diversity synthesis*.

Referring to Figure 3.2, once the packet detection block has indicated the start of a packet, control logic stores the samples corresponding to the preamble's long training symbol S_0 (Figure 3.3) from the upper set of antennas into the first half of a circular buffer entry. Next, the control logic toggles the *AntSel* (for antenna select) line in Figure 3.2, switching to the lower set of antennas for the duration of the second long training symbol S_1.[1] Since S_0 and S_1 are identical and each 3.2 μs long, they fall well within the coherence time[2] of the indoor wireless channel, and so ArrayTrack can treat the information obtained from each set of antennas as if the two sets were obtained simultaneously from different radios at the AP.

[1] ArrayTrack uses the long training symbols because the hardware radio platform has a 500 ns switching time during which the received signal is highly distorted and unusable.

[2] The time span over which the channel can be considered stationary. Coherence time is mainly a function of the RF carrier frequency and speed of motion of the transmitter, receiver, and any nearby objects.

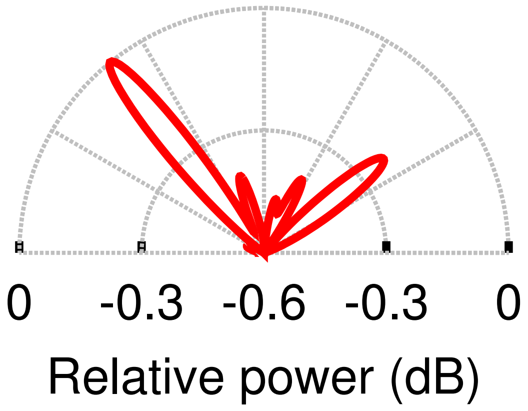


Figure 3.4: The *AoA spectrum* of a client's received signal at a multi-antenna access point estimates the incoming signal's power as a function of its angle of arrival.

3.1.3 AoA spectrum generation

Especially in indoor wireless channels, RF signals reflect off objects in the environment, resulting in multiple copies of the signal arriving at the access point: this phenomenon is known as multipath propagation. An AoA spectrum of a client's received signal at a multi-antenna AP is an estimate of the incoming signal's power as a function of AoA, as shown in Figure 3.4. Since strong multipath propagation is present indoors, the direct-path signal may be significantly weaker than the reflected-path signals, or may even be undetectable. In these situations, the highest peak on the AoA spectrum would correspond to a reflected path instead of the direct path to the client. This makes indoor localization using AoA spectra alone highly inaccurate, necessitating the remaining steps in ArrayTrack's processing chain.

3.1.3.1 Phased-array primer

In order to explain how ArrayTrack generates AoA spectra, I now present a brief primer on phased arrays. While the technology is well established, I focus on indoor applications, highlighting the resulting complexities.

For clarity of exposition, I first describe how an AP can compute AoA information in free space (*i.e.*, in the absence of multipath reflections), and then generalize the principles to handle multipath wireless propagation. The key to computing a wireless signal's AoA is to analyze received *phase* at the AP, a quantity that progresses linearly from zero to 2π every RF wavelength λ along the path from client to access point, as

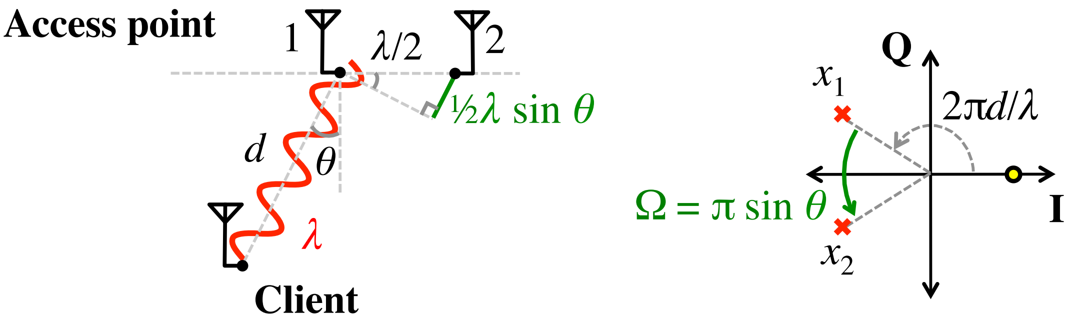


Figure 3.5: Principle of operation for ArrayTrack's AoA spectrum computation phase. (*Left:*) The phase of the signal goes through a 2π cycle every radio wavelength λ, and the distance differential between the client and successive antennas on the access point is related to the client's bearing on the access point. (*Right:*) The complex representation of the sent signal at the client (filled dot) and received signals at the access point (crosses) reflects this relationship.

shown in Figure 3.5 (*left*).

This means that when the client sends a signal, the AP receives it with a phase determined by the path length d from the client. Phase is particularly easy to measure at the physical layer, because software-defined and hardware radios represent the phase of the wireless signal graphically using an *in-phase-quadrature* (I-Q) plot, as shown in Figure 3.5 (*right*), where angle measured from the I axis indicates phase. Using the I-Q plot, we see that a distance d adds a phase of $2\pi d/\lambda$ as shown by the angle measured from the I axis to the cross labeled x_1 (representing the signal received at antenna one). Since there is a $\lambda/2$ separation between the two antennas, the distance along a path arriving at bearing θ is a fraction of a wavelength greater to the second antenna than it is to the first, that fraction depending on θ. Assuming $d \gg \lambda/2$, the added distance is $\frac{1}{2}\lambda \sin\theta$.

These facts suggest a particularly simple way to compute θ at a two-antenna access point in the absence of multipath: measure x_1 and x_2 directly, compute the phase of each ($\angle x_1$ and $\angle x_2$), then solve for θ as

$$\theta = \arcsin\left(\frac{\angle x_2 - \angle x_1}{\pi}\right) \tag{3.1}$$

Generalizing to multiple antennas. In indoor multipath environments, Equation 3.1 quickly breaks down, because multiple paths' signals sum in the I-Q plot. However, adding multiple, say M, antennas can help. The best known algorithms are based on eigenstructure analysis of an $M \times M$ *correlation matrix* $\mathbf{R_{xx}}$ in which the entry at the l^{th} column and m^{th} row is the mean correlation between the l^{th} and m^{th} antennas' signals.

Suppose D signals $s_1(t),\ldots,s_D(t)$ arrive from bearings $\theta_1,\ldots,\theta_D$ at $M > D$ antennas, and that $x_j(t)$ is the received signal at j^{th} antenna element at time t. Recalling the relationship between measured phase differences and AP bearing discussed above, the *array steering vector* is used $\mathbf{a}(\theta)$ to characterize how much added phase (relative to the first antenna) seen at each of the other antennas, as a function of the incoming signal's bearing. For a linear array,

$$\mathbf{a}(\theta) = \exp\left(\frac{-j2\pi d}{\lambda}\right)\begin{bmatrix} 1 \\ \exp(-j\pi\sin\theta) \\ \exp(-j2\pi\sin\theta) \\ \vdots \\ \exp(-j(M-1)\pi\sin\theta) \end{bmatrix} \tag{3.2}$$

d is the separation between antennas which is usually half wavelength λ.

So what the AP hears can be expressed as

$$\mathbf{x}(t) = \overbrace{[\mathbf{a}(\theta_1)\ \mathbf{a}(\theta_2)\ \cdots\ \mathbf{a}(\theta_D)]}^{\mathbf{A}}\begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_D(t) \end{bmatrix} + n(k), \tag{3.3}$$

where $\mathbf{n}(k)$ is noise with zero mean and σ_n^2 variance. This means that $\mathbf{R_{xx}}$ can be expressed as

$$\begin{aligned} \mathbf{R_{xx}} &= \mathbb{E}[\mathbf{xx}^*] \\ &= \mathbb{E}\left[(\mathbf{As}+\mathbf{n})\left(\mathbf{s}^*\mathbf{A}^*+\mathbf{n}^*\right)\right] \\ &= \mathbf{A}\mathbb{E}\left[\mathbf{ss}^*\right]\mathbf{A}^*+\mathbb{E}\left[\mathbf{nn}^*\right] \\ &= \mathbf{AR_{ss}A}^*+\sigma_n^2\mathbf{I} \end{aligned} \tag{3.4}$$

where $\mathbf{R_{ss}} = \mathbb{E}\left[\mathbf{ss}^*\right]$ is the source correlation matrix.

The array correlation matrix $\mathbf{R_{xx}}$ has M eigenvalues $\lambda_1,\ldots,\lambda_M$ associated respectively with M eigenvectors $\mathbf{E} = [\mathbf{e}_1\ \mathbf{e}_2\ \cdots\ \mathbf{e}_M]$. If the noise is weaker than the incoming signals, then when the eigenvalues are sorted in non-decreasing order, the smallest

$M - D$ correspond to the noise while the next D correspond to the D incoming signals. The D value depends on how many eigenvalues are considered big enough to be signals. ArrayTrack chooses D value as how many eigenvalues are larger than a threshold that is a fraction of the largest eigenvalue. Based on this process, the corresponding eigenvectors in $\mathbf{E}$ can be classified as noise or signal:

$$\mathbf{E} = \left[\overbrace{e_1 \, \dots \, e_{M-D}}^{\mathbf{E}_N} \; \overbrace{e_{M-D+1} \, \dots \, e_M}^{\mathbf{E}_S} \right] \tag{3.5}$$

I refer to $\mathbf{E}_N$ as the *noise subspace* and $\mathbf{E}_S$ as the *signal subspace*.

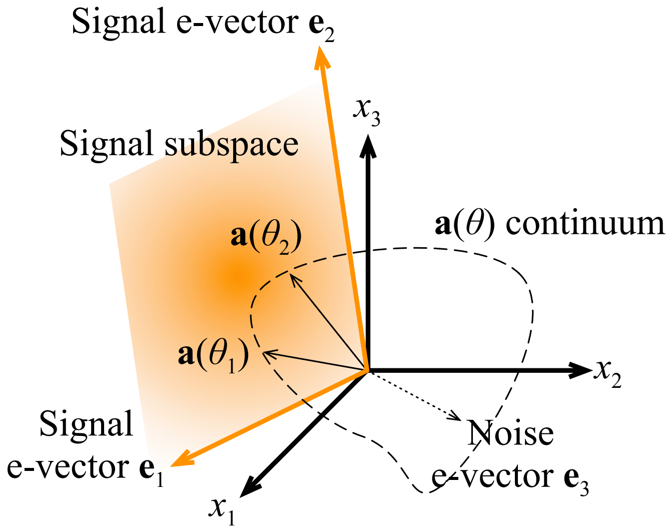

Signal e-vector $\mathbf{e}_2$
Signal subspace
x_3
$\mathbf{a}(\theta_2)$
$\mathbf{a}(\theta)$ continuum
$\mathbf{a}(\theta_1)$
x_2
Signal e-vector $\mathbf{e}_1$
Noise e-vector $\mathbf{e}_3$
x_1

Figure 3.6: In this three-antenna example, the two incoming signals (at bearings θ_1 and θ_2 respectively) lie in a three-dimensional space. Eigenvector analysis identifies the two-dimensional *signal subspace* shown, and MUSIC traces along the array steering vector continuum measuring the distance to the signal subspace. Figure adapted from Schmidt [92].

The MUSIC AoA spectrum [92] inverts the distance between a point moving along the array steering vector continuum and $\mathbf{E}_S$, as shown in Figure 3.6:

$$P(\theta) = \frac{1}{\mathbf{a}(\theta)^* \mathbf{E}_N \mathbf{E}_N^* \mathbf{a}(\theta)} \tag{3.6}$$

This yields sharp peaks in $P(\theta)$ at the signals' AoA.

3.1.3.2 Spatial smoothing for multipath distortion

Implementing MUSIC as-is, however, yields highly distorted AoA spectra, for the following reason. When the incoming signals are phase-synchronized with each other

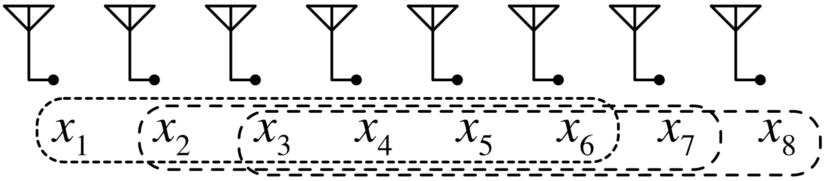


Figure 3.7: Spatial smoothing an eight-antenna array $x_1, \ldots, x_8$ to a virtual six-element array (number of groups $N_G = 3$).

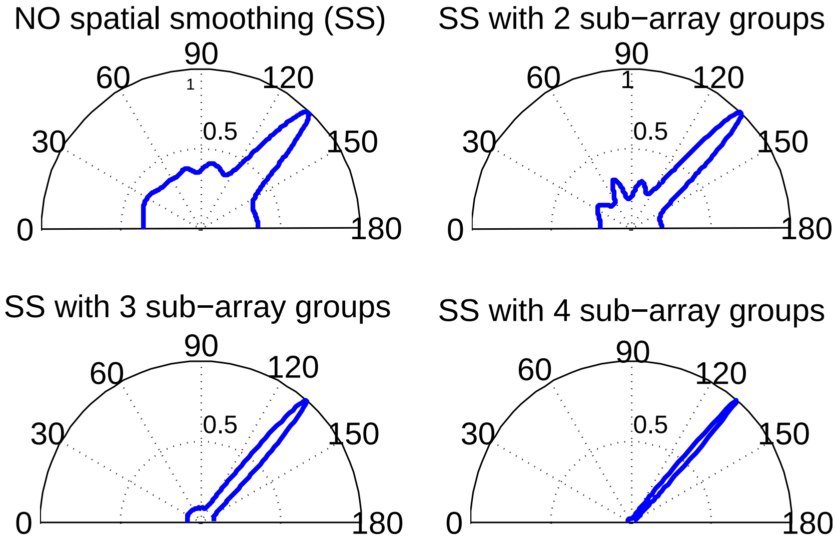


Figure 3.8: Varying the amount of spatial smoothing on AoA spectra.

(as results from multipath) MUSIC perceives the distinct incoming signals as one superposed signal, resulting in false peaks in $P(\theta)$. ArrayTrack therefore adopts *spatial smoothing* [97], averaging the correlation matrices of incoming signals across N_G groups of antennas to reduce this correlation. For example, spatial smoothing over $N_G = 3$ six-antenna groups on an eight-antenna array with correlation matrices R_{16}, R_{27} and R_{38} would output one spatial smoothed correlation matrix: $\dot{R}$, where $\dot{R} = \frac{1}{3}(R_{16} + R_{27} + R_{38})$, as shown in Figure 3.7.

How should ArrayTrack choose N_G? Figure 3.8 shows a microbenchmark computing MUSIC AoA spectra for a client near and in the line of sight of the AP (so the direct-path bearing dominates $P(\theta)$) both with and without spatial smoothing. As N_G increases, the effective number of antennas decreases, and so spatial smoothing can eliminate smaller peaks that may correspond to the direct path. On the other hand, as N_G increases, the overall noise in the AoA spectrum decreases, and some peaks may be narrowed, possibly increasing accuracy. Based on this microbenchmark and experience generating AoA spectra indoors from a number of different clients in my testbed, it's found that a good compromise is to set $N_G = 2$, and this value is used in the performance evaluation in Section 4.3.

3.1.3.3 Array geometry weighting

Information from the linear array is not equally reliable as a function of θ, because of the asymmetric physical geometry of the array. Consequently, after computing a spatially-smoothed MUSIC AoA spectrum, the ArrayTrack multiplies it by a *windowing function* $W(\theta)$, the purpose of which is to weight information from the AoA spectrum in proportion to the confidence that ArrayTrack has in the data. With a linear array, ArrayTrack multiplies $P(\theta)$ by

$$W(\theta) = \begin{cases} 1, & \text{if } 15^\circ < |\theta| < 165^\circ \\ \sin\theta, & \text{otherwise.} \end{cases} \tag{3.7}$$

3.1.3.4 Array symmetry removal

Although a linear antenna array can determine bearing, it cannot determine which side of the array the signal is arriving from. This means that the AoA spectrum is essentially a 180° spectrum mirrored to 360°. When there are many APs cooperating to determine location, this is not a problem, but when there are few APs, accuracy suffers. To address this, ArrayTrack employs the diversity synthesis scheme described in Section 3.1.2 to have a ninth antenna not in the same row as the other eight included. Using the ninth antenna, ArrayTrack calculates the total power on each side of the AoA spectrum, and removes the half with less power, resulting in a true 360° AoA spectrum.

3.1.4 Multipath suppression

While the spatial smoothing algorithms described above (§3.1.3.2) reduce multipath-induced distortion of the AoA spectrum to yield an accurate spectrum, they do not identify the direct path, leaving multipath reflections free to reduce system accuracy. The multipath suppression algorithm I present here has the goal of removing or reducing peaks in the AoA spectrum not associated with the direct path from AP to client.

ArrayTrack's multipath suppression algorithm leverages changes in the wireless channel that occur when the transmitter or objections in the vicinity move by grouping together AoA spectra from multiple frames, if available. The method is motivated

Scenario	Frequency
Direct path same; reflection paths changed	71%
Direct path same; reflection paths same	18%
Direct path changed; reflection paths changed	8%
Direct path changed; reflection paths same	3%

Table 3.1: Peak stability microbenchmark measuring the frequency of the direct and reflection-path peaks changing due to slight movement.

Algorithm (Multipath suppression)

1. Group two to three AoA spectra from frames spaced closer than 100 ms in time; if no such grouping exists for a spectrum, then output that spectrum to the synthesis step (§3.1.5).
2. Arbitrarily choose one AoA spectrum as the *primary*, and remove peaks from the primary not paired with peaks on other AoA spectra.
3. Output the primary to the synthesis step (§3.1.5).

Figure 3.9: ArrayTrack's multipath suppression algorithm.

by the following observation: when there are small movements of the transmitter, the receiver, or objects between the two, the direct-path peak on the AoA spectrum is usually stable while the reflection-path peaks usually change significantly, and these slight movements happen frequently in real life when we hold a mobile handset making calls, for example.

I run a microbenchmark at 100 randomly chosen locations in my testbed (see Figure 3.20, p. 69), generating AoA spectra at each position selected and another position five centimeters away. If the corresponding bearing peaks of the two spectra are within five degrees, ArrayTrack marks that bearing as *unchanged*. If the variation is more than five degrees or the peak vanishes, ArrayTrack marks it *changed*.

The results are shown in Table 3.1. Most of the time, the direct-path peak is unchanged while the reflection-path peaks are changed. This motivates the algorithm shown in Figure 3.9. Note that for those scenarios in which both the direct-path and reflection-path peaks are unchanged, ArrayTrack keeps all of them without any dele-

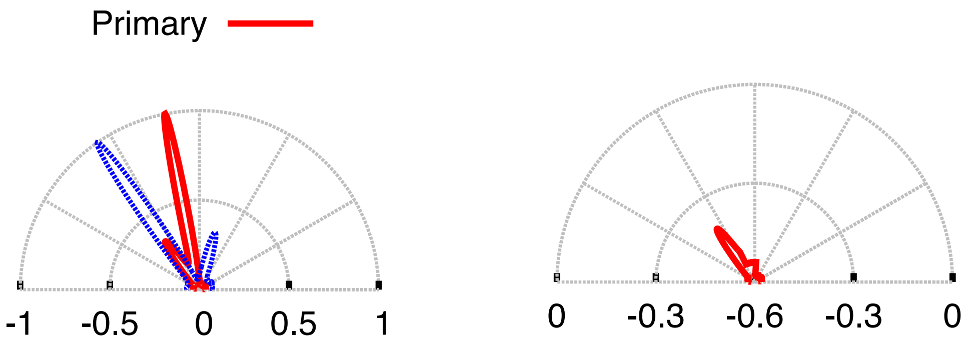


Figure 3.10: ArrayTrack's multipath suppression algorithm operating on two example AoA spectra (*left*) and the output AoA spectrum (*right*).

terious consequences. Also, observe that the microbenchmark above only captures two packets. This leaves room for even further improvement if ArrayTrack captures multiple packets during the course of the mobile's movement. The only scenario which induces failure in the multipath suppression algorithm is when the reflection-path peaks remain unchanged while the direct-path peak is changed. However, as shown above, the chances of this happening are small. An example of the algorithm's operation is shown in Figure 3.10.

3.1.5 AoA spectra synthesis

In this step, ArrayTrack combines the AoA spectra of several APs into a final location estimate. Suppose N APs generate AoA spectra $P_1(\theta), \ldots, P_N(\theta)$ as processed by the previous steps, and ArrayTrack wishes to compute the likelihood of the client being located at position $\mathbf{x}$ as shown in Figure 3.4. ArrayTrack computes the bearing of $\mathbf{x}$ to AP i, θ_i, by trigonometry, and then estimates the likelihood of the client being at location $\mathbf{x}$, $L(\mathbf{x})$, as

$$L(\mathbf{x}) = \prod_{i=1}^{N} P_i(\theta_i). \tag{3.8}$$

With Equation 3.8 ArrayTrack searches for the most likely location of the client by forming a 10 centimeter by 10 centimeter grid, and evaluating $L(\mathbf{x})$ at each point in the grid. ArrayTrack then uses hill climbing on the three positions with highest $L(\mathbf{x})$ in the grid using the gradient defined by Equation 3.8 to refine the location estimate.

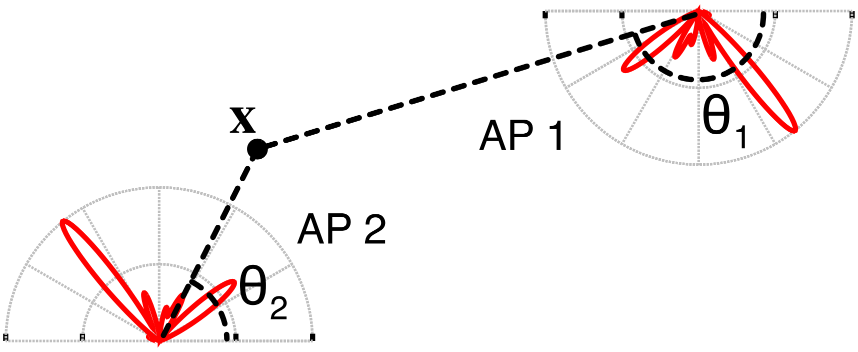


Figure 3.11: ArrayTrack combines information from multiple APs into a likelihood of the client being at location **x** by considering all AoA spectra at their respective bearings (θ_1, θ_2) to **x**.

3.1.6 AoA profile as a unique location signature

AoA information from multiple APs can be combined for localization purpose. On the other hand, it is also noted that a single AoA profile can serve as a unique location signature for security purposes. This signature is very difficult for an attacker to forge while the attacker is not located exactly at the same position as the legitimate client. How a unique AoA signature is generated and compared is described in this section.

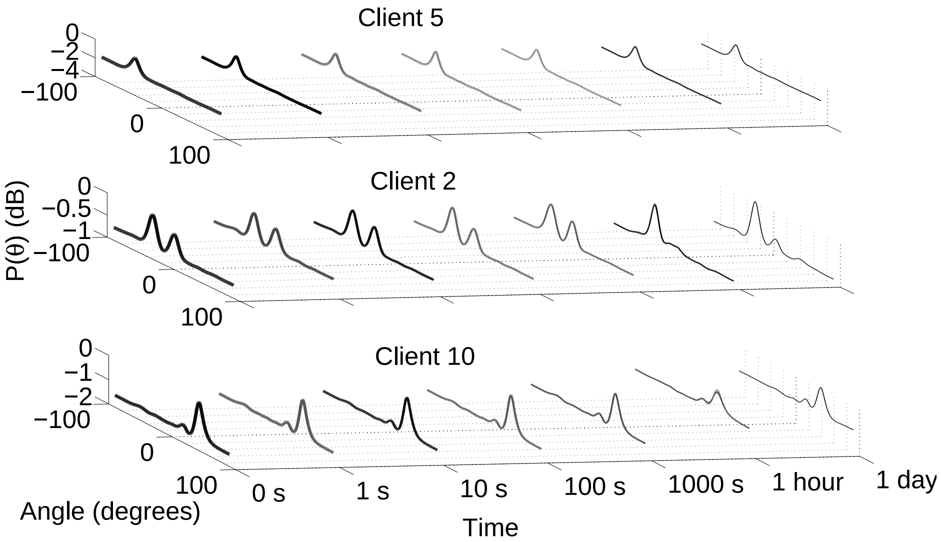


Figure 3.12: Stability of AoA signatures for three clients: each curve on each subplot shows the client's pseudospectrum at logarithmically-spaced time intervals.

To convey the intuition of how the unique AoA spectra are computed, a single client transmitting near an AP with no multipath reflections is considered. If the client is at a bearing θ to the AP as shown in Figure 3.5 (left), then its signal will travel an extra distance of $1/2\lambda \sin\theta$ to the second AP antenna, as compared with the first. This distance directly corresponds to a *measured baseband phase difference* $\Omega = \pi \sin\theta$ between the two signals' baseband symbol representations, as shown in Figure 3.5 (right). Therefore, my estimate of the client's bearing to the AP $\hat{\theta}$ based on a measured baseband phase difference Ω is

$$\hat{\theta} \quad = \quad \arcsin(\Omega/\pi)\,. \qquad (3.9)$$

Prior work has shown that the above concepts generalize to compute AoA spectra using more than two antennas and in the presence of indoor multipath propagation. All that required is a hardware or software-defined radio to capture tens to hundreds of baseband physical layer samples from the preamble of a received packet [124]. It is also noted that the AoA signatures are relatively stable in a static environment. Benchmark experimental results are shown in Figure 3.12. Three clients are placed at three random locations and the AoA pseudospectra are measured at different time points and plotted. We can see clearly from the figure, even after 24 hours, the pseudospectra are still reasonably stable which makes them suitable to be employed for identification purposes.

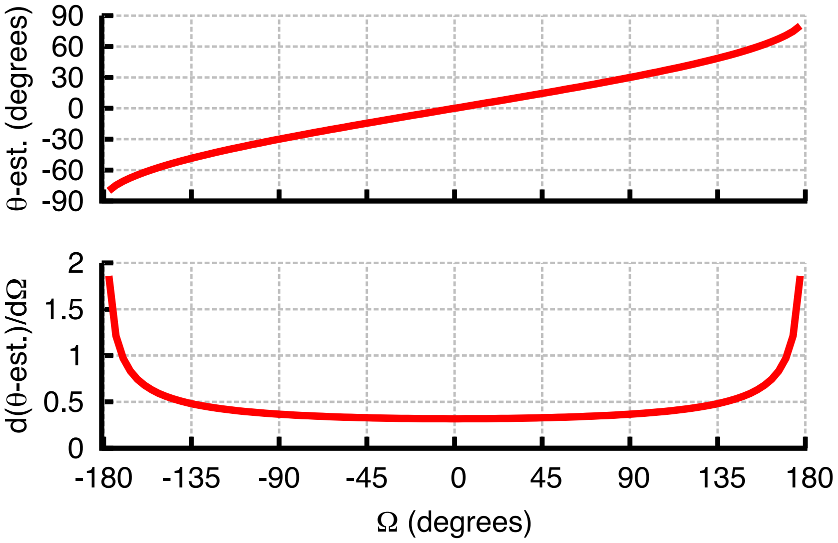


Figure 3.13: *Upper:* Estimated client bearing to the AP $\hat{\theta}$ as a function of measured baseband phase difference Ω, and its rate of change with respect to Ω (*lower*).

It is further noticed that $\hat{\theta}$ and Ω do not have a linear relationship with each other. Estimated client bearing is plotted in Figure 3.13 (top), in units of degrees and the derivation of $\hat{\theta}$ with respect to Ω in n Figure 3.13 (down). Here, note that a small perturbation of Ω translate to smaller perturbations of $\hat{\theta}$ when the client is broadside to the axis of the array (*i.e.*, $\theta \approx 0$) but larger perturbations of $\hat{\theta}$ when the client is close to $\theta = \pm\pi/2$ radians (corresponding to $\Omega = \pm\pi$ radians):

$$\frac{\mathrm{d}}{\mathrm{d}\Omega}\hat{\theta} = \frac{1}{\sqrt{\pi^2 - \Omega^2}} \tag{3.10}$$

$\mathrm{d}\hat{\theta}/\mathrm{d}\Omega$ reaches a minimum of $\pi^{-1} \approx 0.32$ when the client is broadside, and increases sharply when the client is near the array axis.

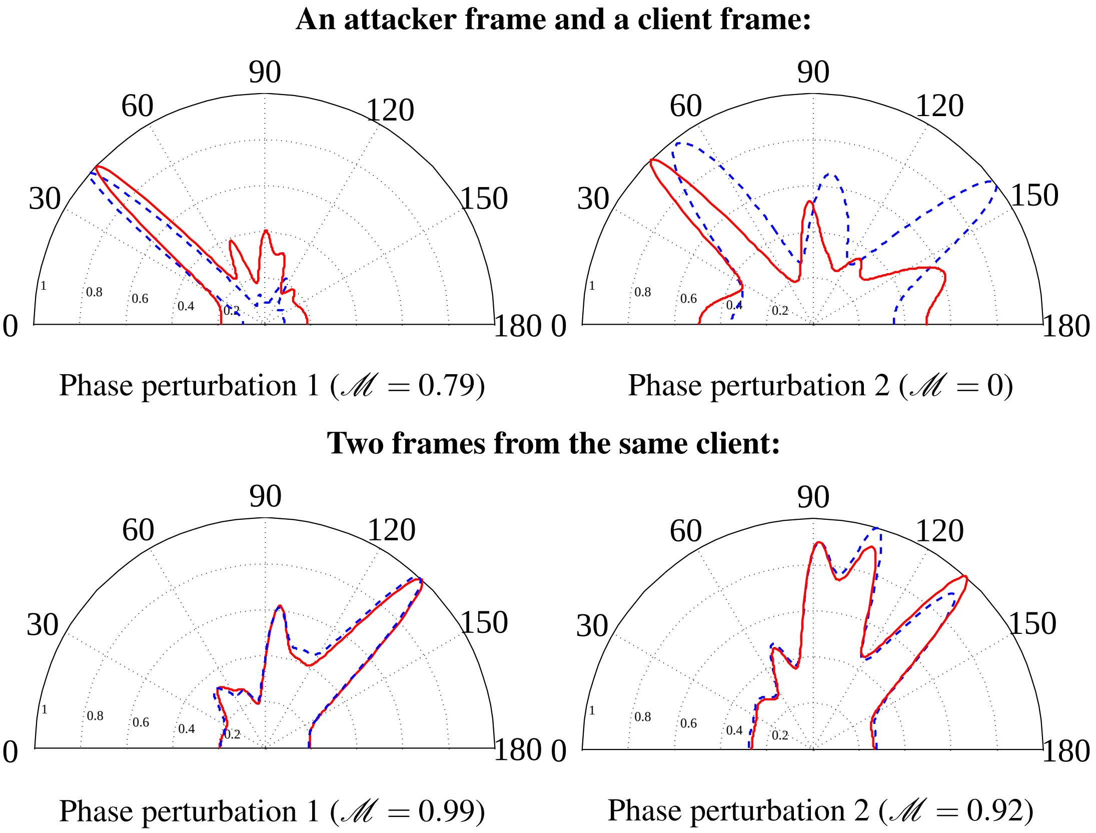
Random phase perturbation. The AoA computation yields one AoA spectrum a time. But since we try to formulate a highly specific client signature, it's preferred to make the array sensitive to slight changes at any angle. Based on the above observation of non-uniform rate of change of $\hat{\theta}$ with respect to Ω, then, I propose a novel scheme to add a random phase offset ζ_2 to Ω, and compute another AoA signature. This process is iterated, adding $L-1$ further random offsets, so obtaining L AoA signatures $\sigma_1(\theta),...,\sigma_L(\theta)$ based on L random phase offsets $\zeta_1 = 0, \zeta_2, \ldots, \zeta_L$. Note that we can introduce deterministic perturbation to bring the peaks close to $\pm\pi$ to make the signature most unique in order to increase the attack detection rate. However, deterministic perturbation has the following disadvantages compared with random perturbations:

1. The false alarm rate will also be increased accordingly.

2. The peaks very close to $\pm\pi$ become unstable (sensitive to very tiny changes).

3. As there are multiple peaks on a signature, it is difficult to bring all the peaks near to $\pm\pi$ at the same time.

A random perturbation scheme is employed to spread the peaks all over the range between $-\pi$ and $+\pi$ to maintain a balance between high detection rate and low false alarm rate. The averaging process described in next section mitigates the inaccuracies caused near +180/-180 degrees and also reduces the possibility of similar lobes in coincidence.



An attacker frame and a client frame:

Phase perturbation 1 ($\mathscr{M} = 0.79$)

Phase perturbation 2 ($\mathscr{M} = 0$)

Two frames from the same client:

Phase perturbation 1 ($\mathscr{M} = 0.99$)

Phase perturbation 2 ($\mathscr{M} = 0.92$)

Figure 3.14: The effect of random phase perturbation on AOA signatures: *Upper:* comparison between an attacker's frame and a client's frame. *Lower:* comparison between two frames from the same client. $\mathscr{M}$ is the similarity metric described below.

Figure 3.14 (upper) shows the effect of my algorithm on two frames transmitted 100 milliseconds apart, from a legitimate client and an attacker placed several centimeters away. The signatures are similar (but not identical) under one phase perturbation, with a rather high similarity metric $\mathscr{M}$), but the other phase perturbation produces very different signatures with a similarity metric $\mathscr{M} = 0$. Figure 3.14 (lower) shows that the random perturbation does not separate (which is desired) the AoA signatures of frames (also spaced 100 ms apart) from the same client, enhancing the AoA signature specificity and selectivity.

Given two AoA signatures $\sigma^{\mathrm{A}}(\theta)$ and $\sigma^{\mathrm{B}}(\theta)$, the local maxima in each signature are found using standard numerical methods. A metric $\mathscr{M}$ is designed that pairs local maximum i from σ^{A} with local maximum j from σ^{B}. The metric takes two pseudospectra σ^{A} and σ^{B} as inputs, and pairs peaks only if they are positioned within a small constant angle threshold Θ of each other. The similarity metric $\mathscr{M}$ also takes the

Speed	Coherence time at 2.4 GHz
Near-stationary (1 kph)	77 ms
Walking (5 kph)	15 ms
Running (12 kph)	6 ms

Table 3.2: Wireless coherence times at 2.4 GHz as computed with Equation 3.12 for typical client motion speeds.

magnitudes (normalized) of paired peaks into consideration. If two peaks are paired in coincidence, their peak magnitudes may vary significantly. For $\mathscr{M}$ to approach 1, the peaks need to be paired and at the same time, the magnitudes of paired peaks should also be close to each other.

$$\begin{aligned} S &= \{(i,j) : |\angle i - \angle j| < \Theta\} \\ \mathscr{M}\left(\sigma^{\mathrm{A}}, \sigma^{\mathrm{B}}\right) &= \frac{\sum_{(i,j)\in S} m_i \cdot m_j}{\left(\sum_i m_i^2 + \sum_j m_j^2\right)/2}. \end{aligned} \tag{3.11}$$

To incorporate the random phase addition algorithm above, the metrics resulting from comparing the L pseudospectra $\sigma_1^{\mathrm{A}}(\theta), ..., \sigma_L^{\mathrm{A}}(\theta)$ arising from random phase perturbations of σ^{A} (as described in the previous section) pairwise with the L pseudospectra $\sigma_1^{\mathrm{B}}(\theta), ..., \sigma_L^{\mathrm{B}}(\theta)$ arising from the same random phase perturbations of σ^{B} are averaged. I call this average metric $\bar{\mathscr{M}}$.

Wireless coherence time. The wireless channel between client and AP is determined by scattering, reflection, and refraction by objects in the environment. A key design parameter is the time duration over which the wireless channel can be considered unchanging with high likelihood, as a function of the speed of the client's motion and/or motion of objects in the environment. The wireless *coherence time* T_c is determined by carrier wavelength λ and maximum client velocity v. If v measured in meters per second, then the coherence time is given by [100]:

$$T_c = \frac{9}{16\pi(v/\lambda)}. \tag{3.12}$$

Table 3.2 shows wireless coherence time at 2.4 GHz as a function of typical client motion speeds indoors. Within this time, we can be confident that the AoA signatures of two transmissions from the same client will indeed be the same, making a false alarm

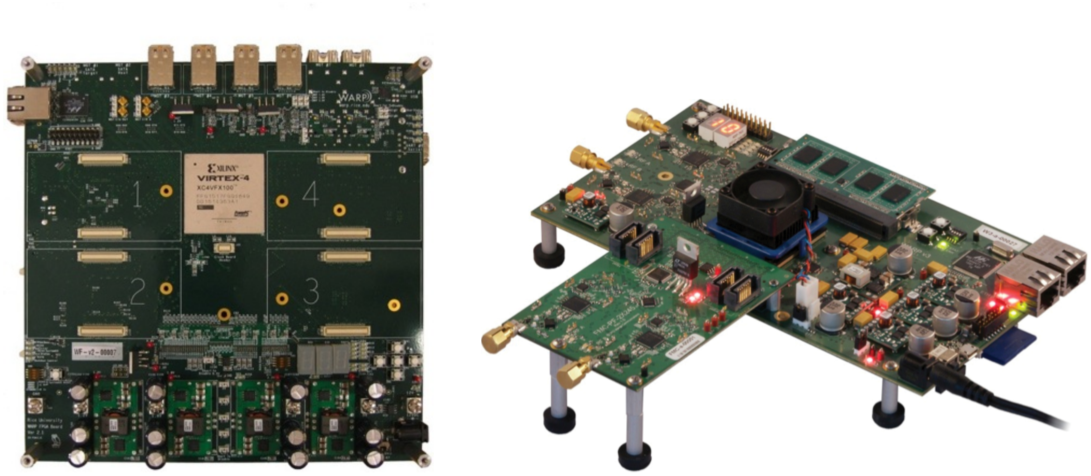


Figure 3.15: The second generation WARP mother board: (*left*) and the third generation WARP board with 4 radio cards attached (*right*) (Figures adopted from Mango website [85]).

a rare event. When the environment is static, the AoA signatures can actually be stable for hours and even days.

3.2 Implementation

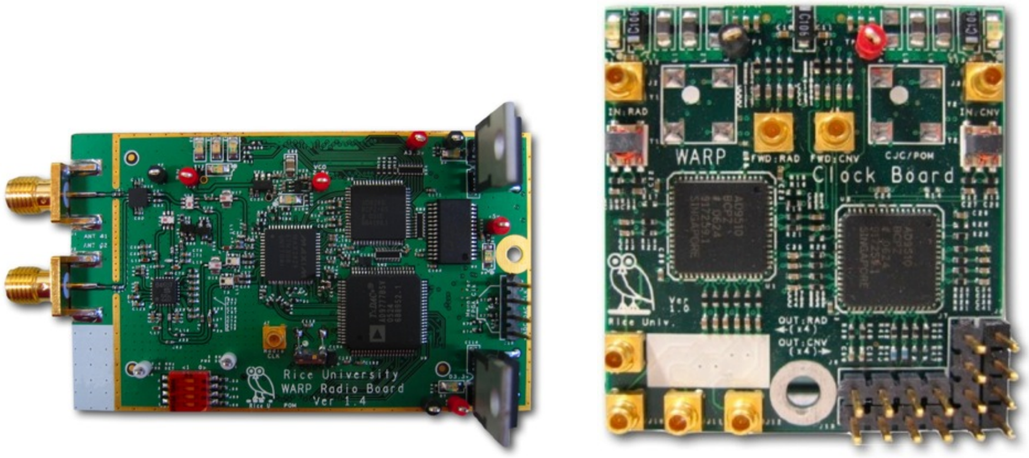


Figure 3.16: The second generation WARP radio board: MAX2829 transceiver (*left*) and the clock board (*right*) (Figures adopted from Mango website [85]).

ArrayTrack is built on top of WARP software-defined radio platform developed by Mango [85]. The second and third generations of WARP boards are shown in Figure 3.15. The second generation mother board has a Xilinx Virtex-4 FPGA [121], which provides all the node's processing resources. Four daughtercards (Wi-Fi radio

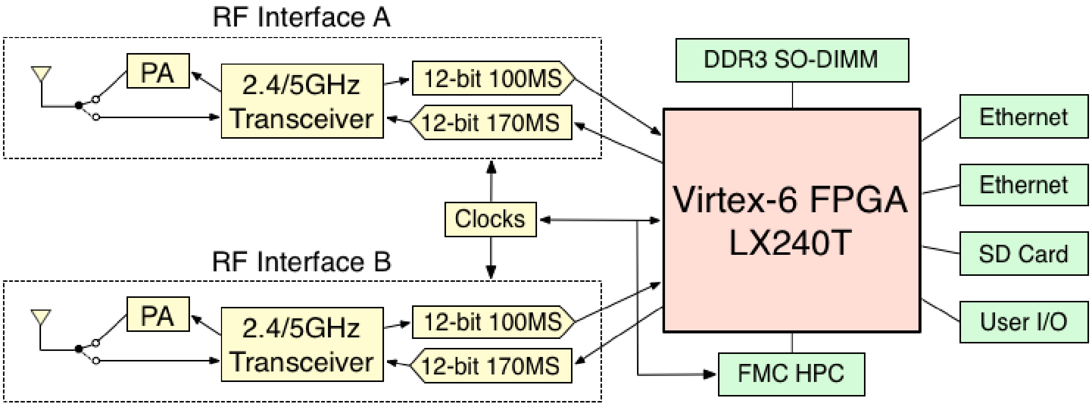


Figure 3.17: The block diagram of third generation WARP platform, integrated with a Virtex-6 FPGA, two programmable RF interfaces (Figure adopted from mango website).

cards) can be attached to the motherboard to support up to 4 radios sharing the same oscillating clock. The Ethernet port connecting WARP and the computer supports both 100 M and Gigabit Ethernet. The default daughtercard is a Maxim MAX2829 [69] transceiver as shown in Figure 3.16 (*left*) supporting both 2.4 GHz and 5 GHz transmission/reception. It supports 14-bit ADC for reception and 16-bit DAC for transmission. The maximum transmission power supported is 18 dBm. The clock board in Figure 3.16 (*right*) is employed to share the sampling clock signal between WARP boards so multiple boards can be time and frequency synchronized. The third generation WARP platform is shown in Figure 3.15 (*right*). Xilinx Virtex-6 LX240T FPGA is employed in this newer version platform for a higher computational speed and two radio boards are already included on the motherboard. The memory slot is also upgraded from DDR2 to DDR3. The maximum radio bandwidth supported is 36 MHz. The block diagram of third generation WARP platform is shown in Figure 3.17. The connections between each components are clearly marked.

3.2.1 ArrayTrack prototype

The prototype ArrayTrack AP, shown in Figure 3.18, uses two Rice WARP FPGA-based wireless radios. Each WARP is equipped with four radio front ends and four omnidirectional antennas. ArrayTrack utilizes the digital I/O pins on one of the WARP boards to output a time synchronization pulse on a wire connected between the two WARPs, so that the second WARP board can record and buffer the same time-indexed samples as the first. The WARPs run a custom FPGA hardware design architected with Xilinx System Generator for DSP that implements all the functionality described in Section 4.1.

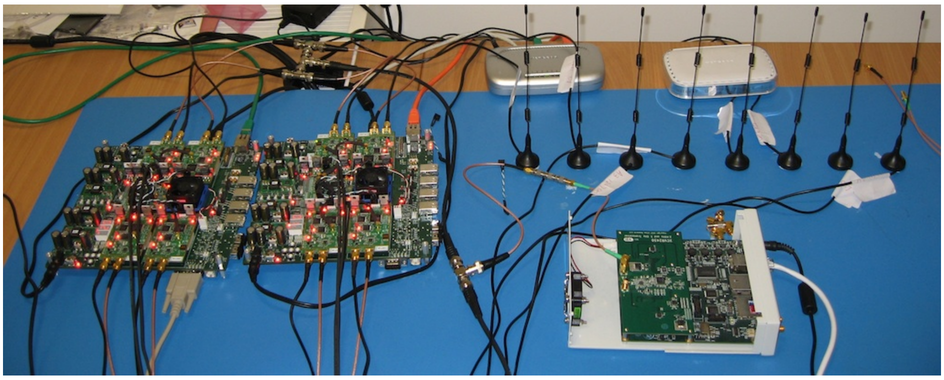


Figure 3.18: the ArrayTrack prototype AP is composed of two WARP radios, while a cable-connected USRP2 software-defined radio calibrates the array.

The 16 antennas[3] attached to the WARP radios are placed in a rectangular geometry (Figure 3.19). Antennas are spaced at a half wavelength distance (6.13 cm). This also happens to yield maximum MIMO wireless capacity, and so is the arrangement preferred in commodity APs.

3.2.2 AP phase calibration

Equipping the AP with multiple antennas is necessary for ArrayTrack, but does not suffice to calculate the AoA as described in the preceding section. Each radio receiver incorporates a 2.4 GHz oscillator whose purpose is to convert the incoming radio frequency signal to its representation in I-Q space shown, for example, in Figure 3.5 (p. 52). An undesirable consequence of this *downconversion* step is that it introduces an unknown phase offset to the resulting signal, rendering AoA inoperable. This is permissible for MIMO, but not for ArrayTrack application, because this manifests as an unknown phase added to the constellation points in Figure 3.5. My solution is to calibrate the array with a USRP2 generating a continuous wave tone, measuring each phase offset directly. Because small manufacturing imperfections exist for SMA splitters and cables labelled the same length, a one-time (run only once for a particular set

[3] The two WAPPs have a total of eight radio boards, each with two ports. ArrayTrack is able to switch ports as described in §3.1.2 and record the two long training symbols with different antennas. So with two WARPs, the maximum number of antennas I can utilize is 16.

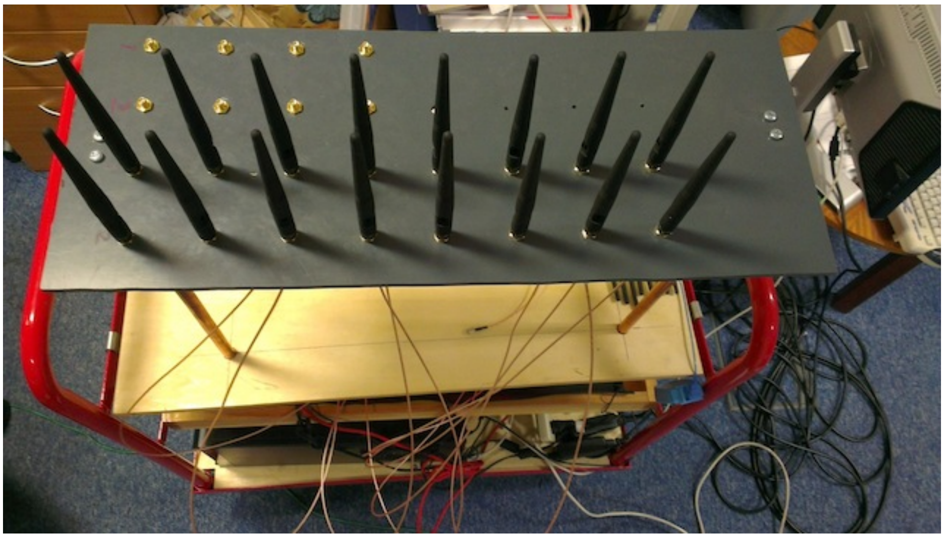


Figure 3.19: The AP mounted on a cart, showing its antenna array.

of hardware) calibration scheme is proposed to handle these equipment imperfections.

The signal from the USRP2 goes through splitters and cables (I called them external paths) before reaching WARP radios. The phase offset Ph_{off} I want to measure is the internal phase difference $Ph_{in2} - Ph_{in1}$. Running calibration once, I obtain the following offset:

$$Ph_{off1} = (Ph_{ex2} + Ph_{in2}) - (Ph_{ex1} + Ph_{in1}) \tag{3.13}$$

Because of equipment imperfections, Ph_{ex2} is slightly different from Ph_{ex1} so Ph_{off1} is not equal to Ph_{off}. The external paths are exchanged and calibration process is run again:

$$Ph_{off2} = (Ph_{ex1} + Ph_{in2}) - (Ph_{ex2} + Ph_{in1}) \tag{3.14}$$

Combing the above two equations, ArrayTrack obtains Ph_{off} and the phase difference caused by equipment imperfections:

$$(Ph_{off2} + Ph_{off1})/2 = Ph_{off} \tag{3.15}$$

$$(Ph_{off2} - Ph_{off1})/2 = Ph_{ex1} - Ph_{ex2} \tag{3.16}$$

Subtracting the measured phase offsets from the incoming signals over the air then cancels the unknown phase difference, and AoA becomes possible.

Testbed clients. The clients used in my experiments are Soekris boxes equipped with Atheros 802.11g radios operating in the 2.4 GHz band.

3.3 Evaluation

To show how well ArrayTrack performs in real indoor environment, I present experimental results from the testbed described in Section 4.2. First I present the accuracy level ArrayTrack achieves in the challenging indoor office environment and explore the effects of number of antennas and number of APs on the performance of ArrayTrack. After that, I demonstrate that ArrayTrack is robust against different transmitter/receiver heights and different antenna orientations between clients and APs. Finally the latency introduced by ArrayTrack, which is a critical factor for a real-time system is examined.

Experimental methodology. The prototype APs are placed at the locations marked "1"–"6" in the testbed floorplan, shown in Figure 3.20. The layout shows the basic structure of the office but does not include the numerous cubicle walls also present. The 41 clients are placed roughly uniformly over the floorplan of size 25 $m \times 24$ m, covering areas both near to, and far away from the AP. Some clients are put near metal, wood, glass and plastic walls to make my experiments more comprehensive. Some clients are placed behind concrete pillars deliberately so that the direct path between the AP and client is blocked, making the situation more challenging.

To measure ground truth in the location experiments presented in this section, ArrayTrack used scaled architectural drawings of the building combined with measurements taken from a Fluke 416D laser distance measurement device, which has an accuracy of 1.5 mm over 60 m.

Due to budget constraints, ArrayTrack used one WARP AP, moving it between the different locations marked on the map in Figure 3.20 and receiving new packets to emulate many APs receiving a transmission simultaneously. This setup does not favor my evaluation of ArrayTrack. Consequently, the results reported next overestimate the magnitude of the location error that ArrayTrack has described, by the following reasoning. The time interval between moving the WARP AP from location to location and taking measurements was approximately minutes, well larger than the wireless channel coherence time, or the time it takes for the wireless channel to change because of motion of objects nearby. Assuming that the nearby object motion between my mea-

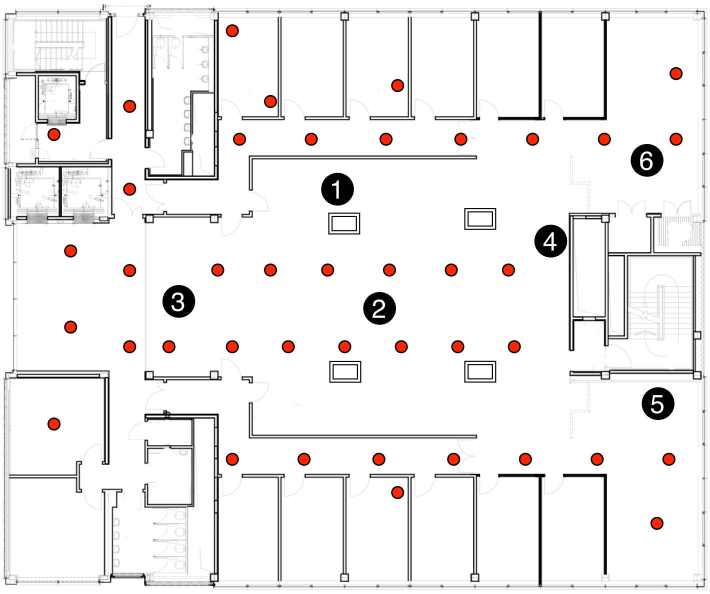


Figure 3.20: Testbed environment: Soekris clients are marked as small dots, and the AP locations are labelled “1”–“6”.

surements at different APs is random and uncorrelated with the wireless channel, such random changes in the wireless channel can only on average add error to ArrayTrack system.

3.3.1 Static localization accuracy

I first evaluate how accurately AoA pseudospectrum computation without array geometry weighting and reflection path removing localizes clients. This represents the performance ArrayTrack would obtain in a static environment without any client movement, or movement nearby. The curves labeled three APs, four APs, five APs, and six APs in Figure 3.21 show raw location error computed with Equation 3.8 across all different AP combinations and all 41 clients. The general trend is that average error decreases with an increasing number of APs. The median error varies from 75 cm for three APs to 26 cm for six APs. The average error varies from 317 cm for three APs to 38 cm for six APs. A heatmap combination example is shown in Figure 3.22 with increasing number of APs.

3.3.2 Semi-static localization accuracy

I now evaluate ArrayTrack using data that incorporates small (less than 10 cm) movements of the clients, with two more such location samples per client. This is represen-

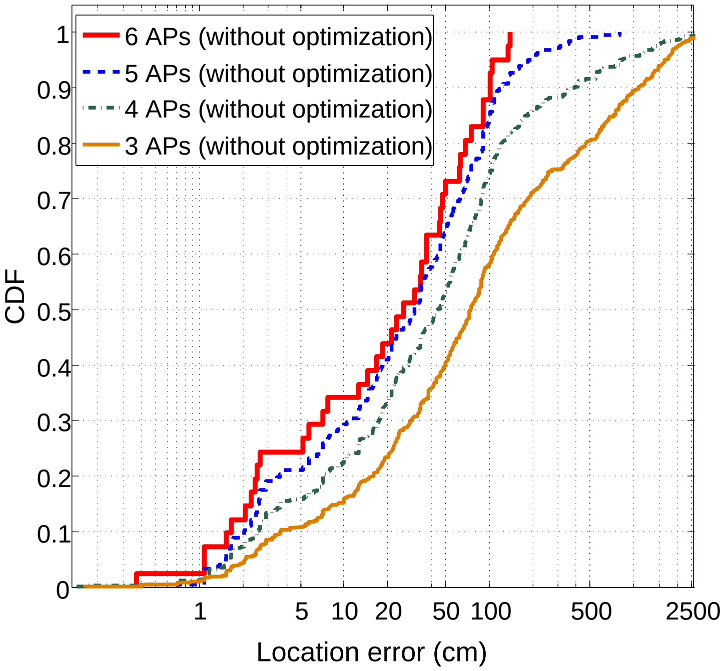


Figure 3.21: Cumulative distribution of location error from unoptimized raw AoA spectra data across clients using measurements taken at all combinations of three, four, five, and six APs.

tative of human movement even when stationary, due to small inadvertent movements, and covers all cases where there is even more movement up to walking speed. In Figure 3.23, it is shown that ArrayTrack improves the accuracy level greatly, especially when the number of APs is small. ArrayTrack improves mean (not median) accuracy level from 38 cm to 31 cm for six APs (a 20% improvement). 90%, 95% and 98% of clients are measured to be within 80 cm, 90 cm and 102 cm respectively of their actual positions. This improvement is mainly due to the array geometry weighting, which removes the relatively inaccurate parts of the spectrum approaching 0 degrees or 180 degree (close to the line of the antenna array).

When there are only three APs, ArrayTrack improves the mean accuracy level from 317 cm to 107 cm, which is around a 200% improvement. The intuition behind this large performance improvement is the effective removal of the false positive locations caused by multipath reflections and redundant symmetrical bearings. When the number of APs is big such as five or six, heatmap combination inherently reinforces the true location and removes false positive locations. However, when the number of APs is small, this reinforcement is not always strong and sometimes the array symmetry

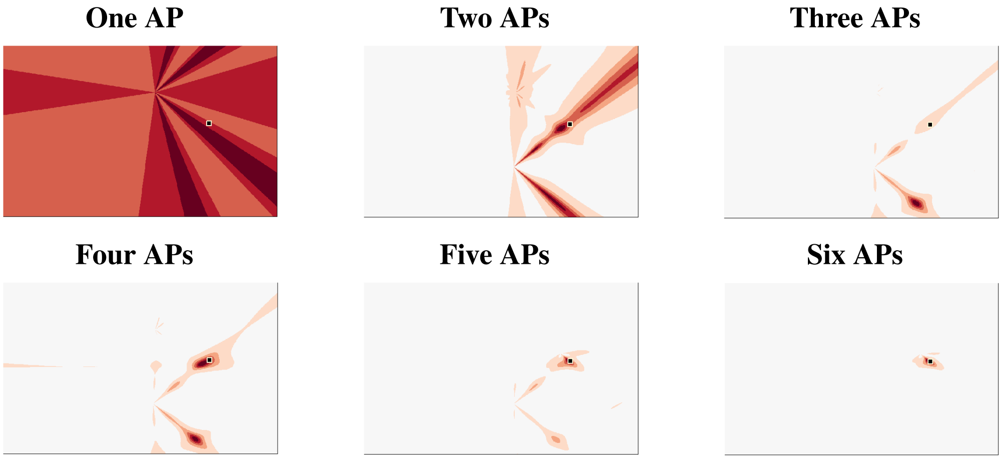


Figure 3.22: Heatmaps showing the location likelihood of a client with differing numbers of APs computing its location. The ground truth location of the client in each is denoted by a small dot in each heatmap.

causes false positive locations, which greatly degrades the localization performance. In these cases, ArrayTrack enables the array symmetry removal scheme described in Section 3.1.3.4 to significantly enhance accuracy. By using this technique, ArrayTrack can achieve a median 57 cm accuracy levels with only three APs, good enough for many indoor applications.

The angle (bearing) accuracy is shown here in Figure 3.24. For clear visualization, only the bearing accuracy of 10 Soekris locations are shown, with 10 measurements at each location. The red dots indicate the ground truth angle while the blue stars are the measured angles at each location. ArrayTrack is able to achieve an average of 3-4 degrees of bearing accuracy with 8-antenna AP.

3.3.2.1 Varying number of AP antennas

I now show how ArrayTrack performs with differing number of antennas at APs. In general, with more antennas at each AP, ArrayTrack can achieve a more accurate AoA spectrum and capture a higher number of reflection-path bearings as Figure 3.25 shows, which accordingly increases localization accuracy. Because ArrayTrack applies spatial smoothing on top of the MUSIC algorithm, the effective number of antennas is actually reduced and so MUSIC algorithm is not able to capture all the arriving signals when the number of antennas is small. The localization performance is shown in Figure 3.26.

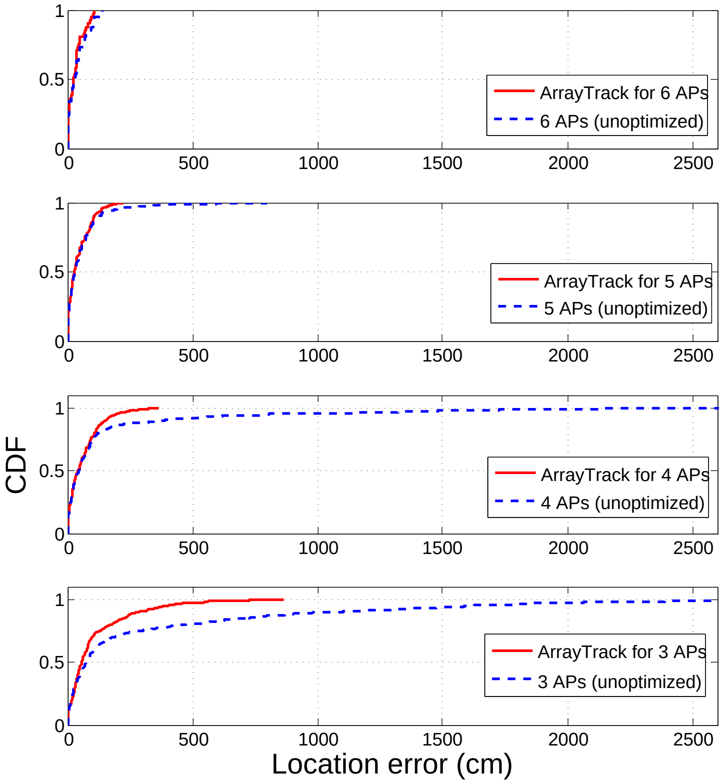


Figure 3.23: Cumulative distribution of location error across clients for three, four, five and six APs with ArrayTrack.

The mean accuracy level is 138 cm for four antennas, 60 cm for six antennas and 31 cm for eight antennas. It is interesting to note that the improvement gap between four and six antennas is bigger than that between six to eight antennas. In a strong multipath indoor environment like my office, the direct path signal is not always the strongest. However, the direct path signal is among the three biggest signals most of the time. How the direct path peak changes is shown in Figure 3.27. The client is placed on the some line with respect to the AP while blocked by more and more pillars. Even when it is blocked by two pillars, the direct path signal is still among the top three biggest, although not the strongest. With five virtual antennas, after spatial smoothing, ArrayTrack is able to avoid losing the direct path signals as sometimes happens when only four antennas are used. The accuracy level improvement from six to eight antennas is due to the more accurate AoA spectrum obtained. With an increasing number of antennas, there will be some point when increasing the number of antennas does not improve accuracy any more as the dominant factor will be the calibration, antenna imperfection, noise, correct alignment of antennas, and even the human measurement

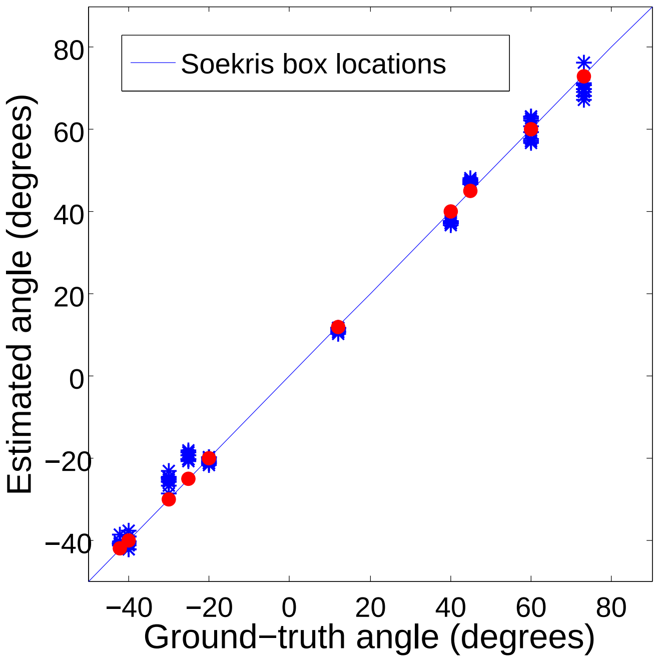


Figure 3.24: Measured bearings versus ground truth bearings for ArrayTrack.

errors introduced with laser meters in the experiments. I expect that an antenna array with six antennas (30.5 cm long) or eight antennas (43 cm long) is quite reasonable.

3.3.3 Robustness

Robustness to varying client height, orientation, low SNR, and collisions is an important characteristic for ArrayTrack to achieve. ArrayTrack's accuracy is investigated under these adverse conditions in this section.

As ArrayTrack works on any part of the packet, the preamble of the packet is chosen to work with ArrayTrack. Preamble part is transmitted at the base rate and what's more, complex conjugate with the known training symbol generate peaks which is very easy to be detected even at low SNR.

3.3.3.1 Height of mobile clients

In reality, most mobiles rest on a table or are held in the hand, so they are often located around 1–1.5 meters off the ground. APs are usually located on the wall near the ceiling, typically 2.5 to 3 meters high. I seek to study whether this height difference between clients and APs will cause significant errors in ArrayTrack's accuracy. The mathematical analysis in §3.1.3 is based on the assumption that clients and APs are at the same height. It is shown below that a 1.5 meter height difference introduces just

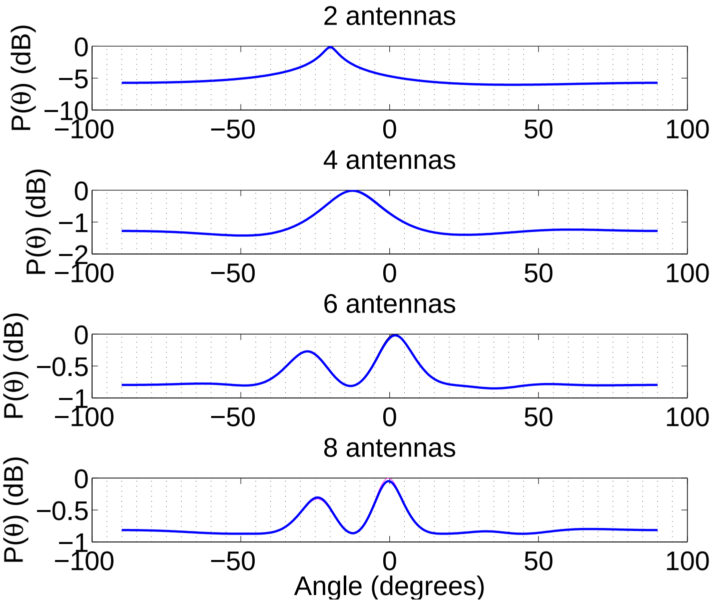


Figure 3.25: More antennas improve resolution and accuracy. Resolution and accuracy benefit localization performance.

1%–4% error when the distance between the client and AP varies between five and 10 meters.

Suppose the AP is distance h above the client; ArrayTrack computes the resulting percentage error caused by this h. AoA relies on the distance difference $d_1 - d_2$ between the client and the two AP antennas in a pair. Given an added height, this difference becomes:

$$d_1' - d_2' = \frac{d_1}{\cos\phi} - \frac{d_2}{\cos\phi} \tag{3.17}$$

where $\cos\phi = h/d$. The percentage error is then $\frac{(d_1'-d_2')-(d_1-d_2)}{d_1-d_2} = (\cos\phi)^{-1} - 1$. For $h = 1.5$ meters and $d = 5$ meters, this is 4% error; for $h = 1.5$ meters and $d = 10$ meters, this is 1% error.

In my experiments, the AP is placed on top of a cart for easy movement with the antennas positioned 1.5 meters above the floor. To emulate a 1.5-meter height difference between AP and clients, the clients are put on the ground at exactly the same location and generate the localization errors with ArrayTrack to compare with the results obtained when they are more or less on the same height with the AP.[4]

The experimental results shown in Figure 3.28 demonstrate the preceding. Me-

[4] Note that this low height does not favor my experimental results as lower AP positions are susceptible to even more clutter from objects than an AP mounted high on the well near the ceiling.

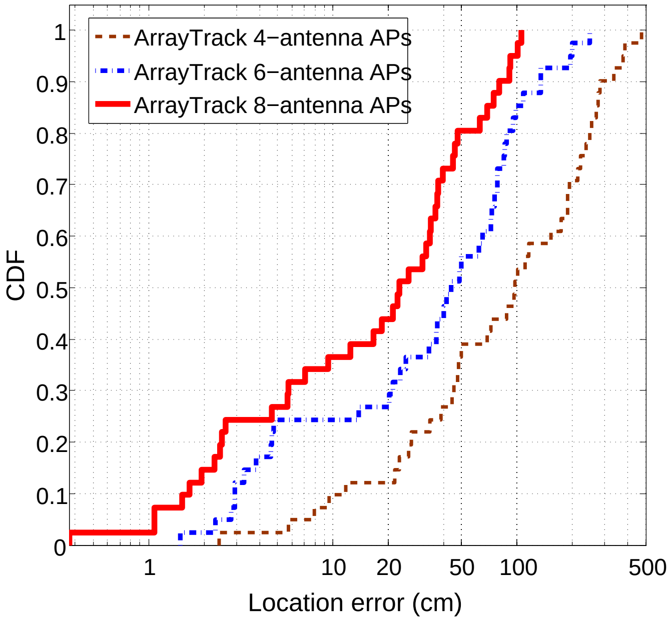


Figure 3.26: CDF plot of location error for four, six and eight antennas with ArrayTrack.

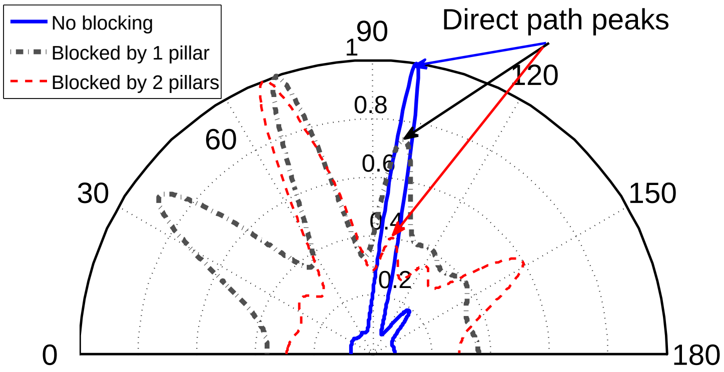


Figure 3.27: The AoA spectra for 3 clients in a line with AP.

dian location error is slightly increased from 23 cm to 26 cm when the AP uses eight antennas. One factor involved is that it is unlikely for a client to be close to all APs, as the APs are separated in space rather than being placed close to each other. One advantage of ArrayTrack is the independence of each AP from the others, *i.e.*, when we have multiple APs, even if one of them is generating inaccurate results, the rest will not be affected and will mitigate the negative effects of the inaccurate AP by reinforcing the correct location.

In future work, I'm planning to extend the ArrayTrack system to three dimensions by using a vertically-oriented antenna array in conjunction with the exiting

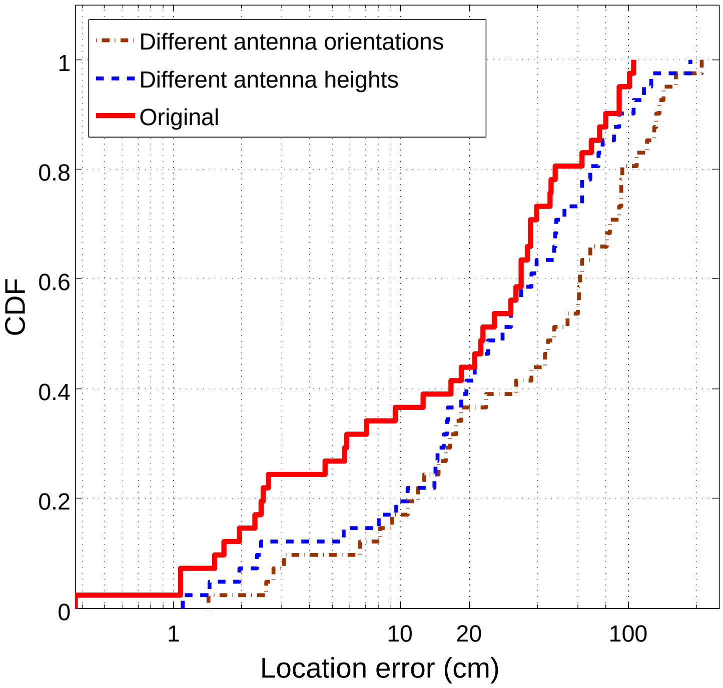


Figure 3.28: CDF plot of ArrayTrack's location error for different antenna height, different orientation and baseline results, with eight antennas and six APs.

horizontally-oriented array. This will allow the system to estimate elevation directly, and largely avoid this source of error entirely.

3.3.3.2 Mobile orientation

Users carry mobile phones in their hands at constantly-changing orientations, so the effect of different antenna orientations on ArrayTrack is studied here. Keeping the transmission power the same on the client side, I rotate the clients' antenna orientations perpendicular to the APs' antennas. The results in Figure 3.28 show that the accuracy level I achieve suffers slightly compared with the original results, median location error increasing from 23 cm to 50 cm. By way of explanation, it is found that the received power at the APs is smaller with the changed antenna orientation, because of the different polarization. With linearly polarized antennas, a misalignment of polarization of 45 degrees will degrade the signal up to 3 dB and a misaligned of 90 degrees causes an attenuation of 20 dB or more. By using circularly-polarized antennas at the AP, this issue can be mitigated.

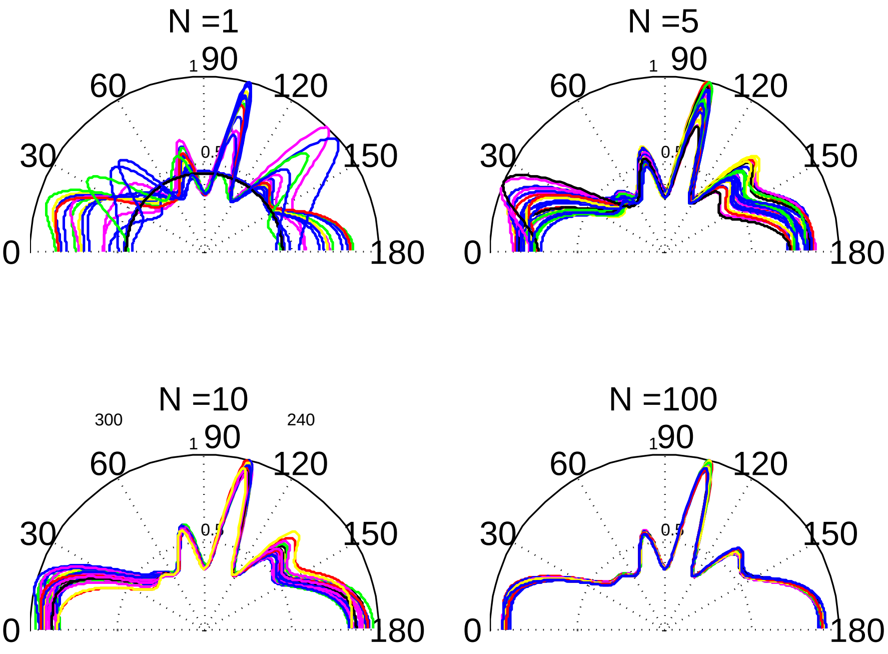


Figure 3.29: The effect of number of data samples on AoA spectrum.

3.3.3.3 Number of preamble samples

To show that ArrayTrack works well with very small number of samples, testbed results are presented in Figure 3.29. Each subplot is composed of 30 AoA spectra from 30 different packets recorded from the same client in a short period of time. I use different number of samples to generate the AoA pseudospectra. As WARPLab samples 40 MHz per second, one sample takes only 0.025 us. When the number of samples increased to 5, the AoA spectrum is already quite stable which demonstrate ArrayTrack has the potential to responds extremely fast. ArrayTrack employs 10 samples in my experiments and for a 100 ms refreshing interval, the overhead introduced by ArrayTrack traffic is as little as: $\frac{(10\text{ samples})(32\text{ bits/sample})(8\text{ radios})}{100\text{ ms}} = 0.0256\,\text{Mbit/s}$.

3.3.3.4 Low signal to noise ratio (SNR)

The signal to noise ratio (SNR) effect on the performance of ArrayTrack is shown in this section. Because ArrayTrack does not need to decode any packet content, all the short and long training symbols can be used for packets detection, which performs very well compared with the original Schmidl-Cox packet detection algorithm. With all the 10 short training symbols used, ArrayTrack is able to detect packets at SNR as low as -10 dB.

It is clear that low SNR is not affecting the packet detection much. Then I want to

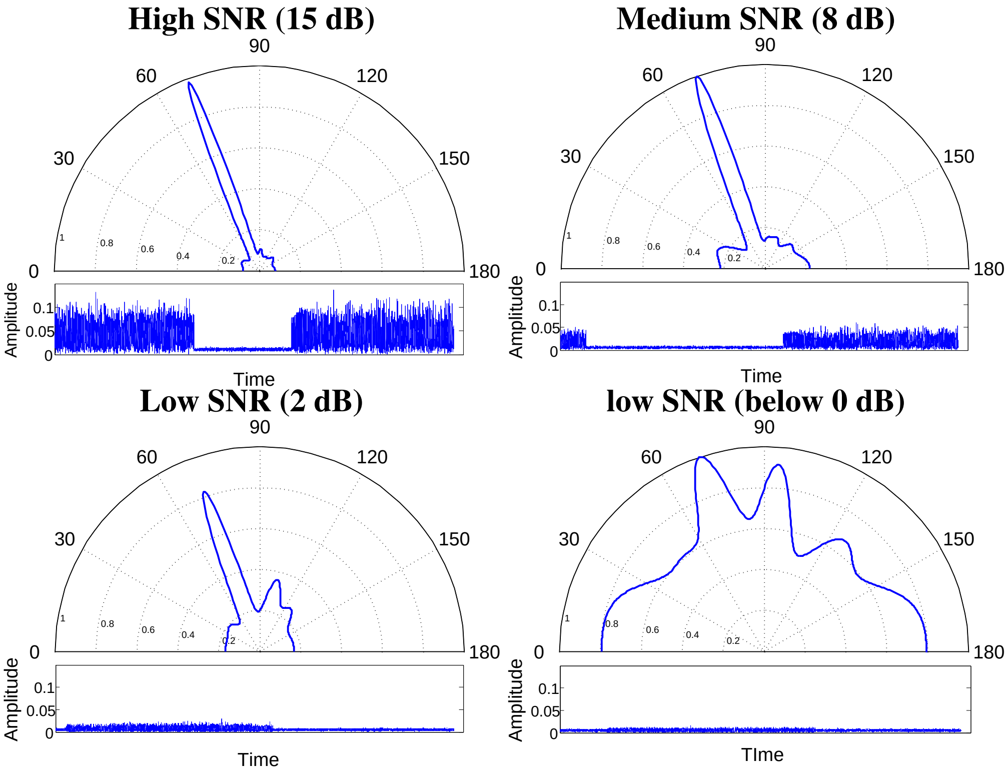


Figure 3.30: AoA spectra become less sharp and more side peaks when the SNR becomes small.

see whether this low SNR affect my AoA performance. The client is kept at the same position untouched and keep decreasing the transmission power of the client to see how AoA spectra change. The results are shown in Figure 3.30. It can be seen clearly that when the SNR becomes very low below 0 dB, the spectrum is not sharp any more and very large side lobe appears on the spectrum generated. This will definitely affect my localization performance. However, it is also found that as long as the SNR is not below 0 dB, ArrayTrack works pretty well.

3.3.3.5 Packet collisions

In very rare cases, two simultaneous transmissions cause collisions. It is shown here ArrayTrack is still able to work as long as the preambles of the two packets are not overlapping. For collision between two packets of 1000 bytes each, the chance of preamble colliding is 0.6%. As long as the training symbols are not overlapping, ArrayTrack is able to obtain the AoA information for both of them. ArrayTrack detects the first colliding packet and generate an AoA spectrum. Then ArrayTrack detects the second colliding packet and generate its AoA spectrum. However, the second AoA spectrum

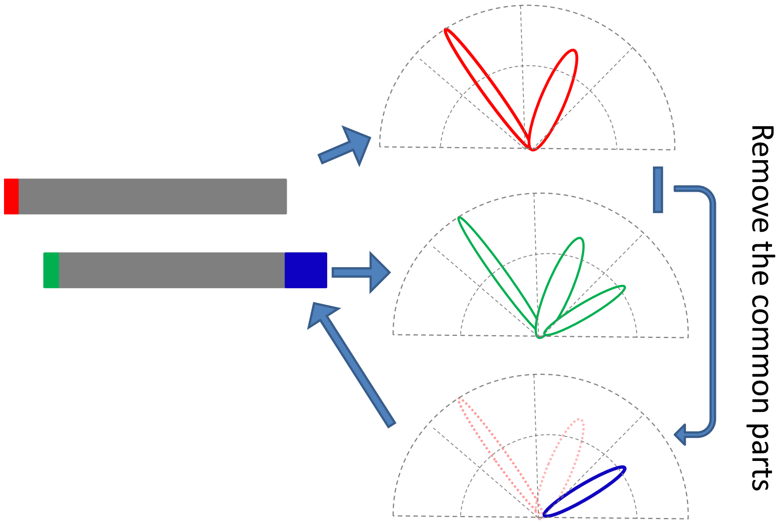


Figure 3.31: The procedure to obtain AoA spectra for two colliding packets.

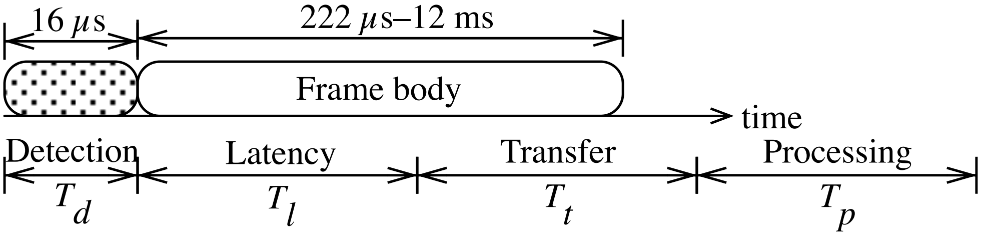


Figure 3.32: A summary of the end-to-end latency that the ArrayTrack system incurs in determining location.

is composed of bearing information for both packets. So ArrayTrack removes the AoA peaks of the first packet from the second AoA spectrum, thus successfully obtaining the AoA information for the second packet as shown in Figure 3.31.

3.3.4 System latency

System latency is important for the real-time applications such as augmented reality. Figure 3.32 summarizes the latency ArrayTrack incurs, starting from the beginning of a frame's preamble as it is received by the ArrayTrack APs. As discussed previously (§3.3.3.3), ArrayTrack only requires 10 samples from the preamble in order to function. ArrayTrack therefore has the opportunity to begin transferring and processing the AoA information while the remainder of the preamble and the body of the packet is still on the air, as shown in the figure. System latency is comprised of the following pieces:

1. T: the air time of a frame. This varies between approximately 222 μs for a

1500 byte frame at 54 Mbit/s to 12 ms for the same size frame at 1 Mbit/s.

2. T_d: the preamble detection time. For the 10 short and two long training symbols in the preamble, this is 16 μs.

3. T_l: WARP-PC latency to transfer samples. It is estimated to be approximately 30 milliseconds, noting that this can be significantly reduced with better bus connectivity such as PCI Express on platforms such as the Sora [103].

4. T_t: WARP-PC serialization time to transfer samples.

5. T_p, the time to process all recorded samples.

T_t is determined by the number of samples transferred from the WARPs to the PC and the transmission speed of the Ethernet connection. The Ethernet link speed between the WARP and PC is 100 Mbit/s. However, due to the very simple IP stack currently implemented on WARP, added overheads mean that the maximum throughput that can be achieved is about 1 Mbit/s. This yields $T_t = \frac{(10 \text{ samples})(32 \text{ bits/sample})(8 \text{ radios})}{1 \text{ Mbit/s}} = 2.56$ ms.

T_p depends on how the MUSIC algorithm is implemented and the computational capability of the ArrayTrack server. For an eight-antenna array, the MUSIC algorithm involves eigenvalue decomposition and matrix multiplications of linear dimension eight. Because of the small size of these matrices, this process is not the limiting factor in the server-side computations. In the synthesis step (§3.1.5) ArrayTrack applies a hill climbing algorithm to find the maximum in the heatmap computed from the AoA spectra. For my current Matlab implementation with an Intel Xeon 2.80 GHz CPU and 4 GB of RAM, the average processing time is 100 ms with a variance of 3 ms for the synthesis step.

Therefore, the total latency that ArrayTrack adds starting from the end of the packet (excluding bus latency) is $T_total = T_d + T_t + T_p \approx 100$ ms.

Recall that the walking speed for human being in indoor environment is around 1-2 m/s. This 100 ms small latency means ArrayTrack is fast enough for locating people with walking speed in the indoor environment.

3.4 Discussion

How does ArrayTrack deal with NLOS?

The NLOS scenarios encountered in the experiments can be categorized into two different groups:

- G1: Direct path signal is attenuated (not the strongest) but exists.
- G2: Direct path signal is totally blocked.

G1 does not affect ArrayTrack as the spectra synthesis method strengthens the true location in nature. Also ArrayTrack proposes the multipath identification scheme to remove multipath reflections when slight movements can be employed.

For G2, one blocked direct path degrades the performance of ArrayTrack slightly but not much. ArrayTrack is dependent on the spectra from multiple APs to localize a client. Because all the APs are placed at different locations, it is very unlikely that the client's direct paths to all the APs are blocked. As long as some of the APs' direct paths are not 100% blocked, ArrayTrack is still able to achieve relatively accurate results. The results presented in this thesis include scenarios when the client is placed with 1-2 walls in between with respect to the APs.

Linear versus circular array arrangement?

Most commonly seen commercial APs have their antennas placed in linear arrangement. As circular array resolves 360 degrees while linear resolves 180 degrees, twice the number of antennas is needed for circular array to achieve the same level of resolution accuracy while linear array has the problem of symmetry ambiguity addressed with synthesis of multiple APs. One big problem with circular array is that the spatial smoothing scheme employed in ArrayTrack can not be applied directly to remove coherence between signals because the steering vector does not possess a linear relationship with circular array.

Chapter 4

Time-Difference of Arrival based localization

Recently, indoor wireless localization systems have broken the meter accuracy barrier both for Wi-Fi devices [47, 124] and RFID tags [112, 113, 130], but to achieve these results, require some combination of many APs and antennas, very long antenna arrays, and/or an RF environment without too many obstacles blocking client-AP lines of sight.

Physical layer	**Bandwidth**	**Raw resolution**
802.11a/g Wi-Fi	20 MHz	15 m
802.11n Wi-Fi	40	7.5
802.11ac Wi-Fi	$<$ 160	$>$ 1.9
Ultra-wideband	$>$ 500	$<$ 60 cm

Table 4.1: Popular physical layers used in localization, their frequency bandwidth, and the raw sample spatial resolution each offers—the distance light travels between sampling instants at that bandwidth: Raw resolution = Speed of light / Bandwidth.

These systems have broken the meter accuracy barrier with AoA and other types of signal processing analysis, but time-of-arrival (ToA) analysis promises to improve accuracy even further. ToA has a particular challenge, however, as shown in Table 4.1: for a typical 802.11a/g Wi-Fi channel with only 20 MHz bandwidth, the signal is sampled once every 50 nanoseconds, during which the signal travels a full 15 meters. As the next rows of the table show, later 802.11n/ac standards enhance this resolution, but still achieve just 1.9 meters of raw sample resolution. Super-resolution spectral signal

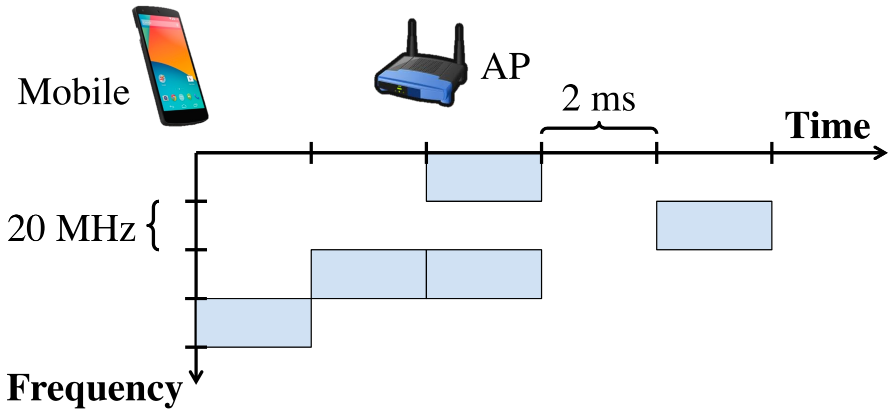


Figure 4.1: A Wi-Fi mobile hops across 80 MHz of bandwidth in 10 ms in order to avoid other competing Wi-Fi users and in-band interference. Cellular LTE mobiles use a similar strategy for similar reasons. ToneTrack leverages in-band frequency hopping to improve indoor localization accuracy.

processing algorithms such as MUSIC [92, 59] and matrix-pencil [89] can enhance this raw sample resolution by an approximate factor of 2×, but still achieve an accuracy proportional to the raw sample spatial resolution shown in Table 4.1, limiting the utility of ToA analysis. Even UWB systems that sample at a rate of 500 MHz and up achieve just 60 cm raw spatial resolution. Focusing on ToA/TDoA analysis, this chapter questions whether it is possible to achieve a higher resolution for time-based localization.

The opportunity leveraged in this work is that tomorrow's wireless networks will make adaptive and opportunistic use of a large variety of frequency bandwidths, ranging from narrow 5 MHz channels intended for the exclusive use of one mobile user at a time, to expansive 160 MHz channels shared between users with CSMA. Indeed, the use of narrow frequency-bandwidth channels is now commonplace: Wi-Fi [15, 20, 101] and cellular systems divide the wireless medium into fine-grained time-frequency blocks, conferring many benefits such as reducing fixed-airtime MAC overheads, increasing SNR, and allowing for channel assignment algorithms to optimize throughput for many users. Furthermore, the use of wide-bandwidth channels has also emerged. The 802.11ac standard [43] specifies transmission bandwidths from 20 to 160 MHz, even allowing two non-contiguous 80 MHz channels to be aggregated together as one 160 MHz channel. Dynamic Frequency Selection (DFS) in the 5.250–5.725 MHz band lets Wi-Fi radios hop channels to avoid any nearby military radar.

In this chapter, I present *ToneTrack*, an indoor localization system that leverages frequency-agile wireless networks to enhance the accuracy of indoor localization. ToneTrack measures the ToA of a client's transmission at pairs of APs in the network. In order to do this, it analyzes the correlation between incoming signals on different subcarriers as MUSIC does, but in the frequency domain. This allows ToneTrack to achieve higher ToA accuracy than simply looking at the sample index of packet detection or channel impulse response. But as noted above, even with super-resolution MUSIC scheme, frequency bandwidth still limits the resolution that ToA algorithms can achieve.

To overcome this bandwidth limitation, ToneTrack contributes a novel signal combination scheme that combines data from a device as it hops across different channels in a frequency band, as shown in Figure 4.1.The result is that ToneTrack can achieve gains in time resolution that are proportional to the number of channels hopped across when transmitting within a channel coherence time.

After extracting a ToA profile of the mobile device's signal from each AP, ToneTrack analyzes each profile individually. Even when multipath reflections arrive too close in time to the direct path and super-resolution schemes reach their resolution limits, failing to resolve all the paths correctly, ToneTrack is still able to identify the useful data therein, retrieving relatively accurate information despite inaccuracy in the overall ToA profile. Here novel peak classification algorithms identify the accurate direct-path peak in the time-of-arrival profile and retain it for further processing.

Lastly, ToneTrack compares TDoA readings across pairs of APs in the network in order to estimate and refine the mobile client's location. Most prior indoor localization work cope with multipath reflections when both reflection paths and direct path exist. The direct path signal may get attenuated but does exist. However, the direct path signal sometimes gets 100% blocked, an even more challenging scenario. ToneTrack employs the classical triangle inequality property to identify the APs whose direct path is completely blocked, improving accuracy in this most challenging situation. Then clustering, outlier rejection, and averaging complete the processing chain, yielding location estimate from a mobile client's transmission. ToneTrack does not require any offline training: preamble data from one to three packets suffice, making the approach amenable to real-time tracking.

Contributions. ToneTrack contributes the following novel design elements:

1. A frequency (tone) combining algorithm that allows a ToA/TDoA method to increase the bandwidth it may utilize for finer accuracy without increasing the radio's sampling rate (§4.1.3).

2. Retrieve useful information from the inaccurate ToA spectrum profile even when the super-resolution scheme reaches the resolution limit (§4.1.4).

3. A triangle inequality-based method together with outlier rejection scheme for identifying and discarding the AP to which a client's LOS transmission is completely blocked (§4.1.5).

Roadmap. The rest of this chapter begins with my system design (§4.1) and implementation (§4.2). My evaluation (§4.3) in an indoor 20×25 meter office testbed demonstrates a 90 cm median localization accuracy with four APs, each equipped with one antenna and overhearing three packets transmitted at adjacent channels with only 20 MHz bandwidth.

4.1 Design

This section presents the design of ToneTrack, starting with a system description (§4.1.1) before delving into ToneTrack's constituent parts: super-resolution ToA processing (§4.1.2), channel combination (§4.1.3), spectrum identification (§4.1.4), and multi-AP data fusion (§4.1.5).

4.1.1 Design Overview

ToneTrack is designed as a passive system that listens to mobile clients' transmissions at nearby APs. Thus the system requires no additional wireless channel overhead for deployment in a production wireless local-area network. Figure 4.2 shows the high-level system design: upon hearing multiple packet transmissions on different channels from a mobile device, an AP forwards the packets to the backend server over a backhaul wired network, appended with timestamps. Then, once the backend server receives this data within a channel coherence time, it passes them to the channel combination step

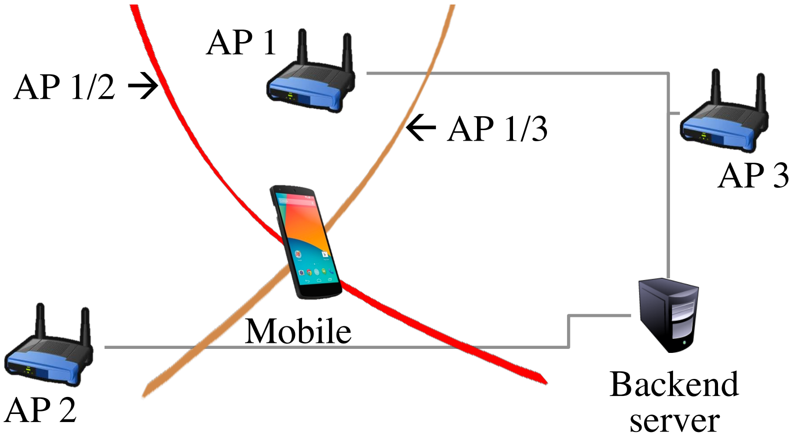


Figure 4.2: *High-level design of ToneTrack.* APs overhear a packet transmission from a mobile and pass the packets to the backend server to run a time-of-arrival (ToA) estimation algorithm, combining the resulting hyperbolic loci (labeled in the figure with their originating AP pairs) for a location estimate.

described in Section 4.1.3 to generate a high-resolution ToA profile. Next, novel algorithms determine whether the resulting ToA profile is in fact accurate, or alternately, contains an accurate part useful for localization, even when the overall ToA profile is inaccurate (§4.1.4). After that, the ToneTrack controller combines the ToA information collected at pairs of APs into TDoA estimates. In Figure 4.2, the hyperbolic curve labeled "AP 1/2" denotes the possible loci of the mobile based on AP 1 and AP 2's TDoA and the hyperbolic curve labeled "AP 1/3" denotes the possible loci of the mobile based on AP 1 and AP 3's TDoA. Finally, the server processes the TDoA estimates across pairs of APs using geometrical reasoning (triangle inequality), clustering and outlier rejection schemes (§4.1.5), yielding a final location estimate.

4.1.2 ToA estimation

Once a client's transmission arrives at an AP, ToneTrack measures the time of arrival (*ToA*) of a client's transmission at one AP: this section describes this process in detail.

4.1.2.1 Primer: MUSIC in the frequency domain

I begin with the classical MUSIC algorithm [92, 59], which models the multipath indoor radio propagation channel $h(t)$ as the sum of D attenuated and delayed impulse responses:

$$h(t) = \sum_{k=1}^{D} \alpha_k \delta(t - \tau_k). \tag{4.1}$$

Here α_k and τ_k are the complex attenuation and propagation delay of the kth path. For simplicity, in this section ToneTrack's operation is described over one Wi-Fi channel. Later sections generalize to multiple Wi-Fi channels.

Processing starts with the per-subcarrier channel response of Equation 4.1 in the frequency domain:

$$\mathbf{H}[f_n] = \sum_{k=1}^{D} \alpha_k e^{-j2\pi(f_0+n\Delta f)\tau_k}. \tag{4.2}$$

Here, f_n and Δf are the carrier frequency and the size of subcarrier bandwidth, respectively. ToneTrack estimates $\mathbf{H}[f_n]$ by taking the DFT of the received 64-sample 802.11 long training symbol and dividing, per-subcarrier, by the known transmitted long training symbol. This estimate is denoted as $\hat{\mathbf{H}}[f_n]$. In 802.11a/g, 52 out of 64 subcarriers contain preamble information; all of them are employed in the processing that follows. The *subcarrier correlation matrix* $\mathbf{R_{HH}}$ then measures phase changes between different subcarriers:

$$\mathbf{R_{HH}} = E\{\hat{\mathbf{H}}[f_n]\hat{\mathbf{H}}^*[f_n]\}, \tag{4.3}$$

where the expectation is calculated across multiple OFDM symbols (spaced in time).

Suppose D copies (direct path and reelection paths) of a transmission $s_1, \ldots, s_D$ arrive at the AP's antenna at D respective times $t_1, \ldots, t_D$, and further suppose the OFDM symbol of the transmission contains M subcarriers ($M > D$) so all copies of the transmission can be captured. Eigenanalysis of the subcarrier correlation matrix $\mathbf{R_{HH}}$ at the AP then results in M eigenvalues associated respectively with M eigenvectors $\mathbf{E} = [\mathbf{e}_1\ \mathbf{e}_2\ \cdots\ \mathbf{e}_M]$. If sorting the eigenvalues in non-decreasing order, the smallest $M - D$ eigenvalues tend to correspond to background noise while the next D eigenvalues tend to correspond to the D incoming copies of the mobile's transmission. Based on this process, the corresponding eigenvectors in $\mathbf{E}$ are classified as noise subspace and signal subspace:

$$\mathbf{E} = \left[\overbrace{e_1 \ldots e_{M-D}}^{\mathbf{E}_N} \overbrace{e_{M-D+1} \cdots e_M}^{\mathbf{E}_S} \right] \tag{4.4}$$

I refer to $\mathbf{E}_N$ as the *noise subspace* and $\mathbf{E}_S$ as the *signal subspace*. The *time steering*

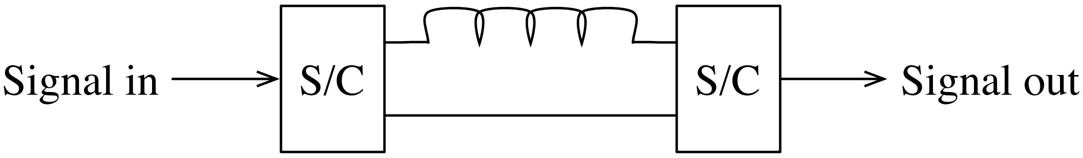


Figure 4.3: A simple two-tap channel emulator: An RF splitter-combiner ("S/C") splits an incoming signals into two branches: one travels over the longer (upper) cabled path, the other travels over the shorter (lower) cabled path. This network models an idealized wireless channel with two paths (one direct path and one reflection path), of varying differential path length.

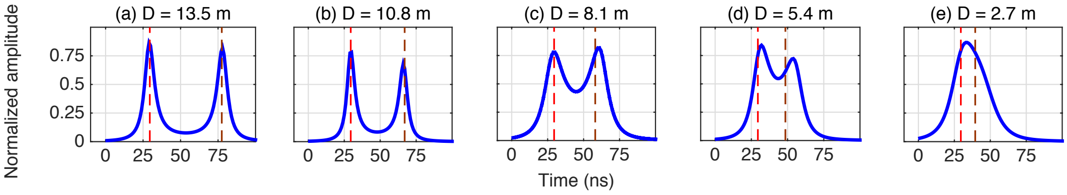


Figure 4.4: *MUSIC's resolution limit.* At 20 MHz bandwidth, MUSIC loses the ability to resolve two paths with a length difference of less than about six meters (20 ft). The two ground-truth path lengths are denoted as dotted vertical lines.

vector $\mathbf{a}(\tau)$ represents the channel's response to a signal arriving at time τ:

$$\mathbf{a}(\tau) = \begin{bmatrix} 1 \\ \exp(-j2\pi\tau\Delta f) \\ \vdots \\ \exp\left(-j2(M-1)\pi\tau\Delta f\right) \end{bmatrix} \tag{4.5}$$

The *time steering vector* $\mathbf{a}(\tau)$ is in the signal subspace and is orthogonal to the noise subspace when τ exactly coincides with each time of arrival of the signal. The *MUSIC ToA spectrum* then measures the distance (in a vector space defined by the array correlation matrix above) between the time steering vector and the noise subspace, as τ varies, thus estimating the time arrival of multiple signals at a granularity independent of the original signal sampling rate:

$$P(\tau) = \frac{1}{\mathbf{a}(\tau)^H \mathbf{E}_N \mathbf{E}_N^H \mathbf{a}(\tau)}. \tag{4.6}$$

With the steering vector and noise subspace vector in the denominator, $P(\tau)$ generates peaks when the steering vector is orthogonal to the noise subspace vector which happens when τ coincides with the time of arrivals of the incoming signals.

Limitations of MUSIC's super-resolution capability. MUSIC is informally known as a *super-resolution* algorithm. The τ variable in Equation 4.6 can vary in arbitrarily-small steps, smaller than the sampling period shown in Table 4.1. But this does not imply MUSIC is able to resolve multipaths with arbitrarily small time delay differences. The frequency bandwidth of the received transmission and background noise imposes a resolution limit independent of τ's step size chosen.

To probe this limit in a controlled experimental setting, the simple channel emulation setup shown in Figure 4.3 is used. An RF splitter-combiner first splits a wired signal into two equal components, one of which travels over a longer cabled path than the other. A second RF splitter-combiner then combines the two signals together, where they are received and processed with MUSIC algorithm. Different cable lengths[1] are used to control the relative path lengths, and attenuators to control the respective path signal strengths to the same level.

Decreasing the path length difference from 13.5 m (44 ft.) gradually to 2.7 m (8.8 ft.) results in the MUSIC pseudospectra shown in Figure 4.4. It can be seen from the figure that MUSIC is able to resolve both paths quite accurately when their lengths are sufficiently different, but once the path length difference between the two signals is decreased to around six meters (20 ft.), MUSIC is not able to generate accurate pseudospectra anymore: its two spectrum peaks half-merge in Figure 4.4 (c) and (d), moving away from ground-truth. When the path length difference is further decreased, the two peaks fully merge into one peak, as shown in Figure 4.4 (e).

4.1.3 Channel combination

To overcome the limitations of MUSIC's super-resolution capability noted above in Section 4.1.2.1, ToneTrack leverages the frequency agility of upcoming Wi-Fi, LTE, and white-space radios as they hop between different frequencies in short periods of time. Note that if frequency hopping happens within a channel coherence time, the ToA spectra generated are similar, but each is a low-resolution picture of the ToA.

[1] Because of lower transmission speed in cable, the cable length is translated to equivalent air propagation distance. (The delay of a 1.8 m RG-58 cable is equivalent to 2.7 m propagation delay in the air).

The basic idea of ToneTrack's channel combination technique is to combine multiple frequency-agile transmissions from the client to form a virtual wider bandwidth transmission, without increasing the sampling rate. Since the effective *array aperture* of MUSIC's ToA estimate is proportional to the number of subcarriers measured (*i.e.*, the bandwidth), time resolution ought to scale linearly with bandwidth. However, naïvely concatenating data from two channels does not work: they must align in both the time and frequency domains in order for the combined data to yield a better resolution in the ToA spectrum plot.

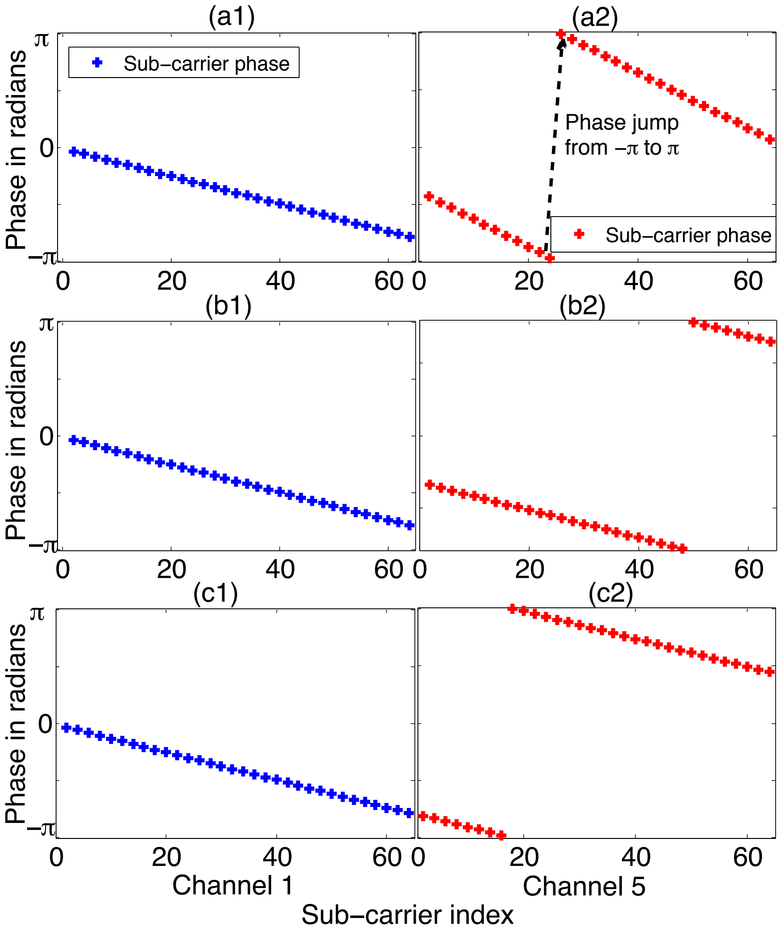


Figure 4.5: *ToneTrack's channel combination scheme.* Time domain alignment equalizes the slope of the phase in the frequency domain between channels, as shown in *(b1)* and *(b2)*. Subsequent frequency domain alignment removes the phase offset and enables successful concatenation of data as shown in *(c1)* and *(c2)*.

Alignment in time domain. While standard packet detection algorithms [91] can synchronize to a sample level at typical baseband sampling rates, ToneTrack requires sub-sample level time alignment of the two overheard signals for combination.

Since the data are recorded at the same radio at different times, there are different fractional (sub-sample) time delays introduced to each set of data. In order to combine data, this random time difference needs to be removed. As these two groups of data are recorded within a small time interval, the relative amplitudes of the peaks on the spectra are stable. Standard fractional interpolation methods [53] are applied to align the two signals based on their respective ToA spectra. Sub-sample interpolation of the raw data in the time domain causes the whole ToA spectrum to move in time. One sample shift is corresponding to a spectrum movement of 50 ns at 20 MHz. The time difference (in ns) of the largest peak position on each of the two spectra is measured. Then the two sets of data in the time domain are aligned, matching the two largest peaks to the same position with a corresponding sub-sample interpolation of the raw data. As demonstrated in Figure 4.5 (b1) and (b2), with a single signal, time domain alignment equalizes the slopes of the two groups of sub-carrier phases.

To see why this is the case, consider a measurement of phase in the frequency domain. Looking across subcarriers of separation Δ_f, the time-shifting property of the DFT

$$H[k]e^{2\pi j\tau_0 k/N} \overset{\mathscr{F}}{\longleftrightarrow} h\left[(n-\tau_0)_N\right] \tag{4.7}$$

tells us that if there is only one signal, phase at the AP changes linearly across subcarriers as $2\pi\Delta_f\tau_0/N$ where the slope of the phase is proportional to the propagation time τ_0.

Alignment in frequency domain. Unfortunately, concatenating even time-aligned data from adjacent channels fails again, yielding completely inaccurate and noisy ToA spectra. ToneTrack needs to estimate the phase of the sub-carrier just after the last sub-carrier of the first channel. Then the phase of the first sub-carrier of the second channel is aligned to the estimated one by subtracting the phase offset. This concept is demonstrated in Figure 4.5 (c1) and (c2) with data from two channels fully aligned in both time and frequency domains. With this step, the two groups of channel response data can now be concatenated to yield better resolution than any one alone. A larger virtual bandwidth is formed without increasing the radio's sampling rate. When multipath is present, the phase change becomes highly non-linear since it is the super-

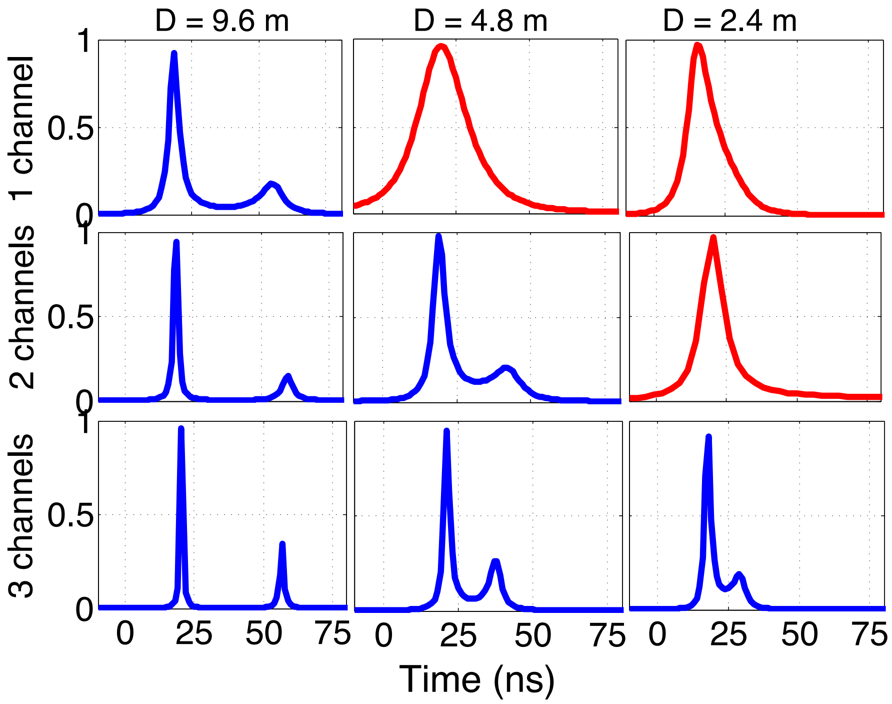


Figure 4.6: ToneTrack's channel combination scheme effectively increases the resolution capability of MUSIC, as tested by varying the path length difference *d* in my two-tap channel emulator. Red curves denote ToA spectra where peaks have problematically merged and MUSIC is not able to resolve them correctly.

position of many paths of varying magnitude and phase. The insight is that since this superposition remains continuous across channels in phase, ToneTrack can still align the two groups of data by matching the phase of the last sub-carriers in the first group of data to the first sub-carrier in the second group of data. With the scheme described here, ToneTrack is able to concatenate multiple groups of data from adjacent channels seamlessly to perform like one single larger bandwidth channel.

4.1.3.1 Channel combining microbenchmark

The effectiveness of ToneTrack's channel combination is demonstrated with the microbenchmark results shown in Figure 4.6. In the first row, with one single 20 MHz channel, ToneTrack fails to resolve both signals when the path difference D between the signals decreases to 4.8 meters (15.8 feet). With the channel combination scheme applied with two channels, ToneTrack successfully resolves both two signals at $D = 4.8$ meters but fails to resolve when the difference decreases to 2.4 m. With three channels, ToneTrack is able to resolve two signals separated by only a 2.4 m (7.9 feet) path length difference. My end-to-end localization results in Section 4.3.2 leverage this channel combining algorithm to markedly improve ToneTrack's accuracy level. The channel combination process at each AP is fully independent of the data fusion process later in Section 4.1.5 across multiple APs.

There is no limit on the number of channels can be employed for combination in ToneTrack. My current implementation is based on 2.4 GHz and thus ToneTrack employs channels 1, 5 and 9 for experiments. The spectrum range available in 2.4 GHz for Wi-Fi is small. Channel 11 may be included to further add 10 MHz to the combination. The spectrum range in 5 GHz is much larger and more channels can be combined for higher accuracy.

4.1.3.2 Overlapping and non-adjacent channels

In the case of overlapping channels that may result when the mobile changes its center frequency by an amount less than the bandwidth of its transmissions, it is clear that ToneTrack's channel combining technique generalizes by averaging the channel information in the tones the two transmissions have in common. Then the two overlapping channels can be converted into two equivalent adjacent channels in terms of localization with the overlapping part removed from one channel.

I briefly discuss how the scheme may be generalized to sets of channels that are non-adjacent. The steering vector needs to be modified to reflect the different subcarrier separation between non-adjacent channels. This has the drawback of multiplying the number of peaks in the ToA spectrum, in a way analogous to the grating lobes problem RF-IDraw solves for AoA spectra [113]. Alignment in the frequency domain is challenging for non-adjacent channels because it is not easy to estimate the correct phase offset, since the phase change is non-linear in the presence of strong multipath. The design and evaluation of non-adjacent channel combination is left as future work.

4.1.4 Spectrum identification

I now describe the processing ToneTrack performs on the ToA profile computed in §4.1.2 and §4.1.3 to determine whether the spectrum is accurate and if not, whether ToneTrack can still retrieve relatively accurate direct-path information from the spectrum. This processing is termed *spectrum identification.* As noted in Section 4.1.2, when the lengths of a line-of-sight path and a reflected path are too close to each other,

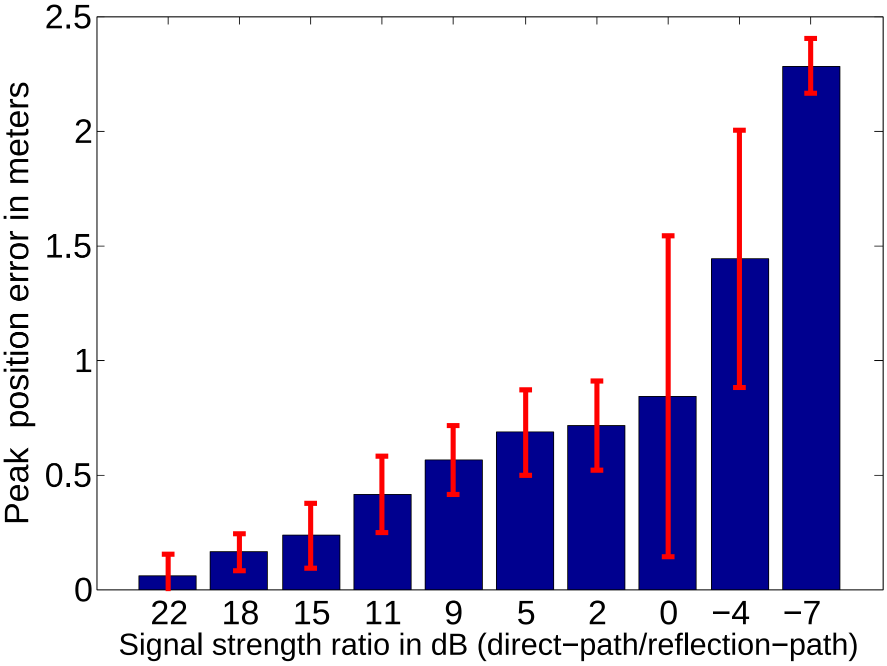


Figure 4.7: Peak position error when two peaks merged into one, as a function of the relative strength of the two peaks.

MUSIC is unable to resolve the two signals correctly in the time domain on the pseudospectrum. This leads to either inaccurate pseudospectrum peak positions or multiple peaks merge. However, ToneTrack can sometimes still retrieve useful and relative accurate information from these *inaccurate* pseudospectra.

4.1.4.1 Merged-signal peaks

It is first observed that when the two paths' peaks merge into one as shown in Figure 4.6, as long as the first (direct) path signal is stronger, the error in the peak position is still small. The simple two-tap channel emulator of Figure 4.3 on page 88 is used to quantify this experimentally. In Figure 4.7, the relative signal strength between the direct path and a reflection path 2.7 meters longer is varied, starting from $+22$ dB (*i.e.*, direct path 22 dB stronger than reflection path) down to -7 dB (*i.e.*, reflection path 7 dB stronger than reflection path). The results in Figure 4.7 show that the error is well under one meter as long as the direct-path signal is stronger. The error increases significantly when the reflection path is stronger, up to 2.3 meters.

After ToneTrack identifies a merged peak, it measures the *skew direction* of the peak as shown in Figure 4.8 by finding the peak position and the two midpoints at which the peak amplitude falls by half (this is also known as the -3 *dB beamwidth*).

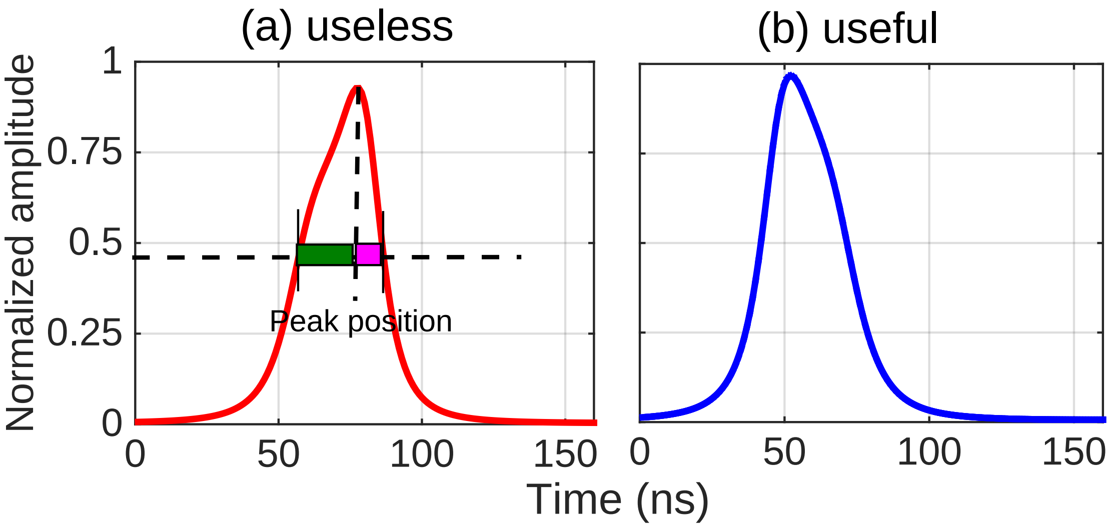


Figure 4.8: Merged-signal peaks. ToneTrack classifies useful spectra by the skew direction (earlier or later) of a merged peak: *(a)* the first (direct path) peak has merged into a later (reflection path) peak, or *(b)* a later (reflection path) peak has merged into the first (direct path) peak.

By comparing the distance of the peak position to the two 3 dB beamwidth midpoints, ToneTrack measures the direction of the peak's skew: a peak position falling to the right of the −3 dB beamwidth's midpoint as shown in Figure 4.8 (a) indicates that the first peak, which corresponds to the direct path, has merged into a later peak (which corresponds to a reflection path). ToneTrack identifies this merged peak as inaccurate and thus useless. The blue plot shows a spectrum skewing earlier in time (merged towards the direct-path peak). In this case the peak has a reasonably small error, and can thus still be kept for localization even it is a peak merged with two signals.

4.1.4.2 Single-signal peaks

If the two peaks are separated by more than the MUSIC resolution limit[2] as shown in Figure 4.10 (a1) and (a2), then MUSIC can accurately estimate their respective positions, and ToneTrack feeds the position of the first, direct-path peak to the next processing stage. But if the two peak positions are separated by less than the resolution limit as shown in Figure 4.10 (b1) and (b2), they fall into the zone that MUSIC is not able to resolve accurately.

[2] It is experimentally verified that at 20 MHz at medium-high SNRs, this resolution limit is stable and measured to be around 6 m.

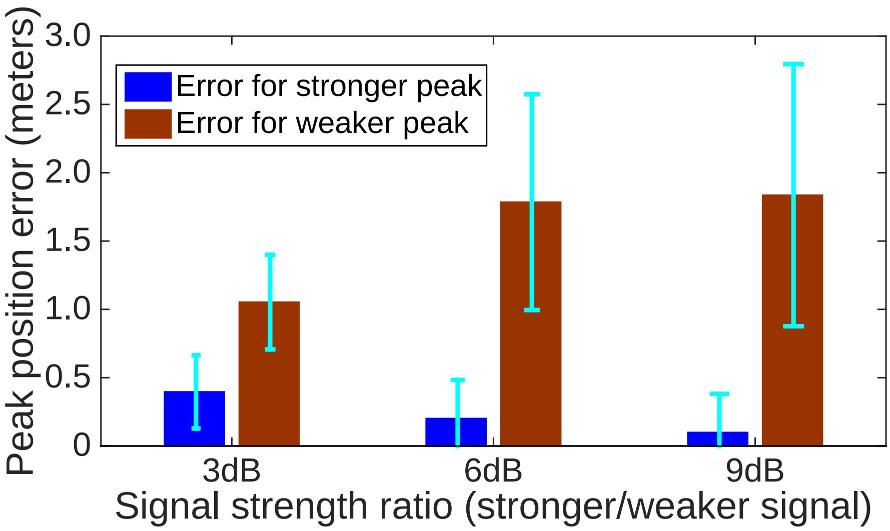


Figure 4.9: Peak error (translated from time to meters) for separated peaks in the simple two-tap channel emulator, when the direct path and the first-arriving reflection path are separated but arrive too close in time for MUSIC to accurately resolve.

Often, the direct path and reflection path signals have differing amplitudes. It is observed anecdotally that even when the two peaks are too close for MUSIC to resolve, the larger peak on the pseudospectrum corresponding to the stronger signal is still quite accurate compared to the smaller peak. This observation is validated empirically in the simple two-tap channel emulator of Figure 4.3 on page 88 with the following microbenchmark. The path length difference between the direct-path and reflection-path signals is fixed to be 5.4 m (18 ft). Then the relative signal strength between the direct and reflection paths is adjusted from 3 dB to 9 dB and the peak position error is shown in Figure 4.9. It can be seen clearly that when the direct path signal is stronger, although MUSIC is not able to resolve both of them correctly, the error of the direct-path peak is quite small (less than 0.5 m). On the other hand, the smaller reflection path peak has a much larger error, so ToneTrack can still extract relatively accurate information from the MUSIC spectrum in these scenarios as ToneTrack only cares the direct-path peak. Referring to Figure 4.10 (b1) and (b2), when the separated peaks are closer than the resolution limit, ToneTrack compares the relative amplitudes of the two. If the amplitude of the first peak is greater than the second peak as in (b2), ToneTrack marks the ToA spectrum as useful; otherwise it discards the ToA spectrum such as in (b1). Referring again to the signals with a 3 dB difference in Figure 4.9, this bounds the error due to

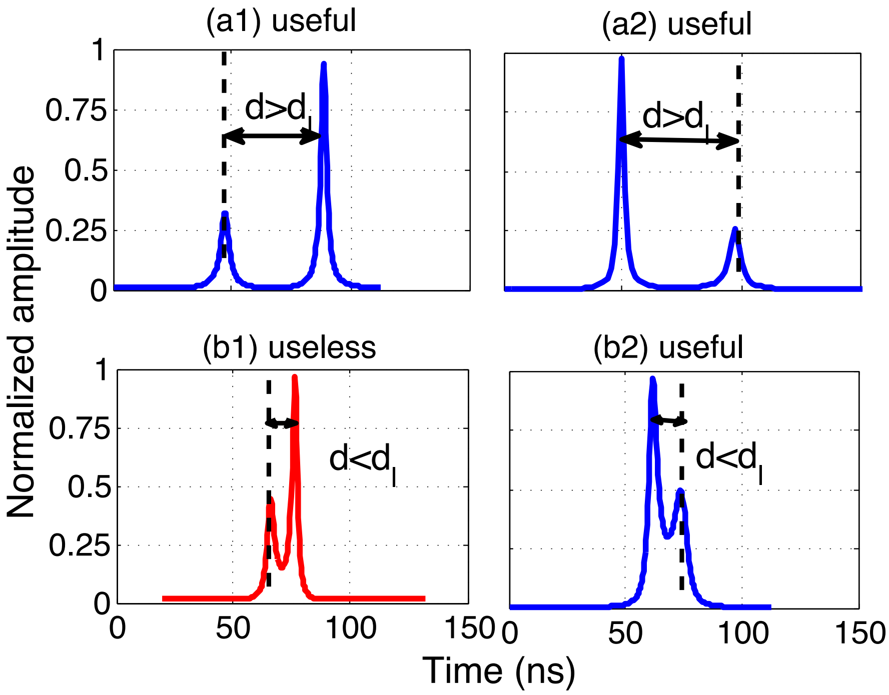


Figure 4.10: The peak separation d is greater than the resolution limit d_l, both (a1) and (a2) are kept. If the d is smaller than d_l, ToneTrack identifies useful ToA spectra by comparing their respective amplitudes and (b1) is discarded.

the presence of the second peak to well below one meter.

4.1.4.3 Classifying peaks as merged or single-signal

To apply the above spectrum identification technique, ToneTrack needs to estimate whether a certain peak arises from a single path or is the result of the respective peaks of multiple arrivals merging together in the ToA spectrum. Prior theoretical work [2] has shown that the beamwidth of the MUSIC spectrum is inversely proportional to the square root of SNR and the bandwidth of the signal. Consequently, within the SNR range ToneTrack operates at, a single-signal peak will be thinner compared to a merged peak, even if the merged peak originates from two closely-spaced signals. For example, the difference is apparent when the red merged peaks in Figure 4.6 is compared with blue peaks. ToneTrack thus measures the peak's −3 dB beamwidth $W_{-3\text{ dB}}$ and compares it with a threshold value W_t to make the decision:

$$W_{-3\text{ dB}} > W_t : \text{Merged peak.}$$

$$W_{-3\text{ dB}} \leq W_t : \text{Single peak.}$$

Using microbenchmarks measuring the impact of SNR and the path difference of the two signals on W_t, ToneTrack experimentally determines the best value for W_t in Section 4.3.3, and show that it produces good end-to-end performance in my indoor testbed in Section 4.3.2.

4.1.4.4 Algorithm (Spectrum Identification)

As the preceding microbenchmarks show, useful and accurate information can still be retrieved even when MUSIC fails to resolve all the signals correctly, as long as information about the direct path peak is relatively accurate. In this section I summarize ToneTrack's *spectrum identification* algorithm, which comprises the processing ToneTrack performs on each (possibly channel-combined, *cf.* §4.1.3) ToA spectrum from a single AP before passing that ToA spectrum on to the multi-AP data fusion step described next in Section 4.1.5.

- **Step 1.** Isolate the first two peaks on the ToA spectrum as input to the algorithm. If the two peak positions are separated by greater than the resolution limit, then the first peak contains accurate direct-path distance information, so ToneTrack retains the spectrum and proceed to Step 3. Otherwise, the two peak positions are separated by a distance less than resolution limit (which MUSIC is not able to resolve accurately) so ToneTrack proceeds to Step 2:

- **Step 2.** Compare the relative amplitudes of the two peaks. From the microbenchmarks, it is known that as long as the direct path signal is stronger than the reflection path signal, the direct-path peak position will be more accurate. So ToneTrack retains the spectrum if and only if the first peak's amplitude exceeds the second's.

- **Step 3.** Check whether the first peak is a single-signal peak or a merged peak (§4.1.4.3). ToneTrack retains the spectrum and the algorithm terminates in this step if the peak is a single-signal peak. Otherwise, ToneTrack proceeds to Step 4:

- **Step 4.** Check the direction of the peak's skew (§4.1.4.1). ToneTrack retains the spectrum if and only if the peak is merged towards the direct path (left side).

After the above steps, only the useful peak remains. At this point ToneTrack sends the ToA spectrum to the multi-AP data fusion step described next.

4.1.5 Multi-AP data fusion

In this final stage of processing, ToneTrack converts measured ToAs from each AP into distance differences between pairs of APs, using these distance differences to estimate the mobile's location. Occasionally, the direct-path signal may be totally blocked, with only reflection signals detectable at the AP. The following two methods are proposed to handle this very challenging scenario.

4.1.5.1 Triangle inequality

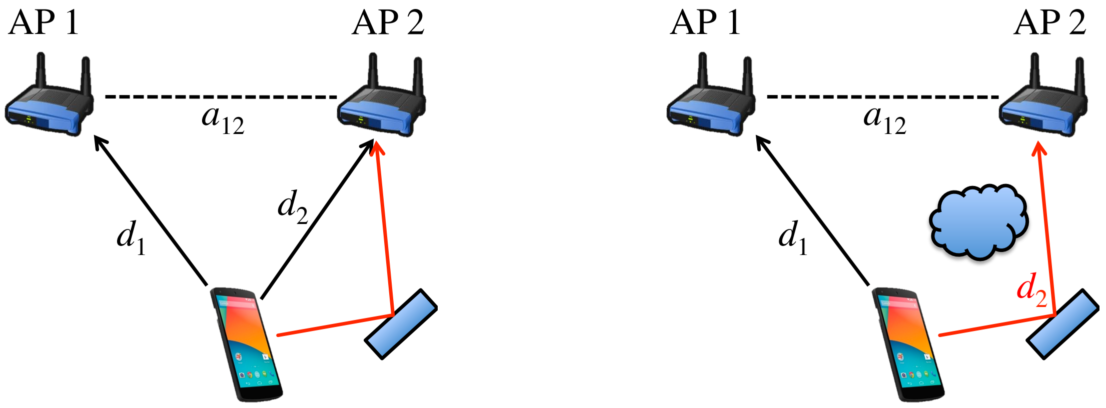
As shown in Figure 4.11a, when both APs are able to resolve the direct-path signals from the mobile client, the distance estimates to AP 1 and AP 2 (d_1 and d_2, respectively), fit the following triangle inequality property:

$$d_1 + a_{12} \geq d_2, \tag{4.8}$$

where a_{12} is the distance between APs 1 and 2, which is known. However, when the direct path to AP 2 is completely blocked and only one or more reflection paths exist, as shown in Figure 4.11b, the resulting distance estimates may violate this triangle inequality, *i.e.*, $d_1 + a_{12} < d_2$. Whenever ToneTrack detects such a violation of the triangle inequality, ToneTrack tags the violating AP (AP 2 in this example) as having its direct path completely blocked, and exclude it from further processing in the chain. It is noted that it is also possible that when the direct path to AP 2 is blocked, the triangle inequality may not necessarily be violated, and so while this test is conservative in the APs it excludes (thus aiding performance), it is not comprehensive in the elimination of direct path blockage scenarios. With more group of APs, the chance of detection of the blocked APs is higher. Also this scheme may fail when multiple APs are 100% blocked. However, the chance that multiple APs are blocked at the same time is quite low as the APs are usually placed at different locations, and my end-to-end evaluation suffers from these effects as and when they happen in practice. It is noted that this method has very recently been applied to ToA-based ultrasound positioning [114] and ToneTrack would like to apply this method to TDoA-based Wi-Fi localization in ToneTrack.

4.1.5.2 Clustering and outlier rejection

Clustering and outlier rejection further reduce the error caused by a complete blockage of the direct path signal and errors from other sources. This is based on the fact that



(a) No △ violation.

(b) △ violation.

Figure 4.11: The classical triangle inequality can identify APs whose direct path to the client must be blocked.

the direct path signals of multiple APs will localize the clients close to the true source location, while reflection path signal will localize the client at random locations. As shown in Figure 4.12, APs 1, 2, 3 and 4 all have direct path signals while AP 5 has direct path signal blocked. Its estimates with any three APs from {1, 2, 3 and 4} will be around the true location of the mobile. A location estimate from involving AP 5 will be far away from the true location, and can be detected and removed. ToneTrack can even detect the AP with direct path totally blocked. Note that ToneTrack need at least four APs whose direct paths are not blocked in order to detect the blocked AP. When the number of available APs is large, the number of combinations ToneTrack needs to check can be very large. One solution to this problem is to remove some APs with small signal strength and only keep the rest for outlier rejection purposes. Please note that it is very unlikely to have the LOS paths to all the APs 100% blocked as the APs are usually placed at different locations. In the extreme scenario when the client is placed in a metal container with LOS paths to all the APs blocked, ToneTrack is not able to localize the client correctly.

4.1.5.3 Final location estimation

As noted above in Section 4.1.1, each pair of APs yields one TDoA estimate in the shape of a hyperbolic arc. Thus three APs are able to localize the client at the inter-

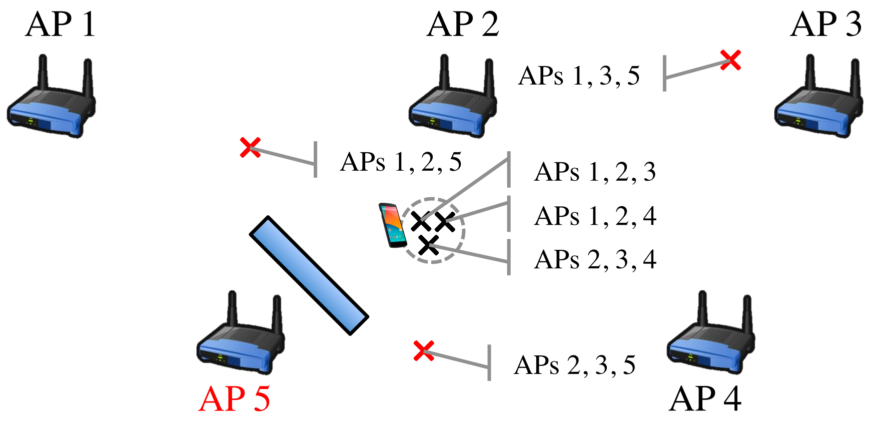


Figure 4.12: Employing clustering and outlier rejection to remove non-accurate estimates.

section of two hyperbolas.[3] Both closed form solutions and iterative algorithms can be found in [14, 70, 93]. ToneTrack leverages a closed form solution in 2-D space similar to prior work [70]. With any group of three APs, ToneTrack has one intersection from two hyperbolas. If there are more than four APs, ToneTrack applies the scheme described in Section 4.1.5.1 and Section 4.1.5.2 to detect the 100% blocked AP and remove it from localization. Then ToneTrack averages the location estimates with all combination of three APs. When only four APs exist, the scheme described in Section 4.1.5.2 can not be applied. ToneTrack then adopts a simple clustering algorithm to choose a group of three estimates which yields the minimum sum of distances and average them.

From ToA to TDoA. ToneTrack is based on time-difference-of-arrival (TDoA) between the mobile transmission's arrival at each pair of APs. In order to compute TDoA, ToneTrack relies on a time-synchronization mechanism between APs. This is achieved by either a wireless protocol such as SourceSync, which can achieve 5 -10 ns (95^{th} percentile) synchronization error at a typical wireless SNR ratio of 20 dB [80], or the Ethernet-based Precision Time Protocol standardized as IEEE 1588, which Broadcom has shown can provide a five nanosecond time synchronization error [11]. Other schemes include time-synchronization with light [60] and the use of distributed antenna system (DAS) [127] to bypass this time synchronization problem.

The computational load of ToneTrack is mainly a matrix multiplication of size

[3] If there is no intersection, ToneTrack discards data from that triplet of APs.

64 x 64. It is noted that with channels combined together, the matrix size is increased linearly. When there are many channels, it is recommended to select one sub-carrier out of every *N* adjacent sub-carriers evenly to reduce the matrix size. When there are many APs, the number of combinations for outlier rejection scheme is large which impedes ToneTrack's real-time objective. ToneTrack thus only keeps a limited number of APs based on the signal strengths as higher SNR presents us more accurate spectrum.

4.2 Implementation

ToneTrack is implemented on the Rice WARP platform [85] with WARPLab version 7.3. ToneTrack employs a small part of the preamble of a packet which is the most robust part for my localization. For the long training symbol (LTS) in the preamble, only the middle 52 out of 64 sub-carriers are actually used. With the original LTS, only 52/64 $\times$ 20 MHz = 16.25 MHz bandwidth would be used for localization. In order to use all subcarriers, ToneTrack builds one symbol very similar to the LTS in 802.11 but with all the 64 subcarriers occupied. This symbol is attached just after the original LTS, incurring less than 0.1% overhead in a 1500-byte packet.

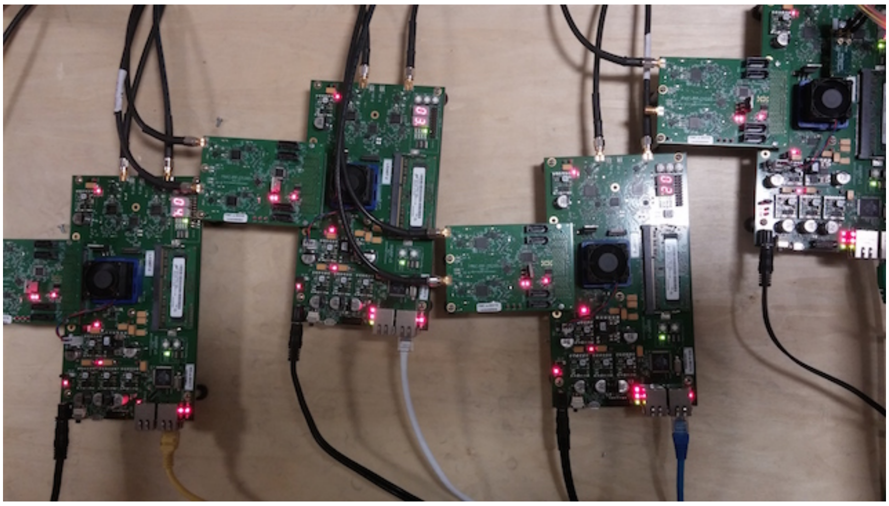


Figure 4.13: Each AP is a latest WARP v3 Kit with FMC-RF-2X245 module to enable 4 radios. Antennas are placed at the dedicated AP positions with low loss LMR-400 cables.

ToneTrack employs five WARPs, one as the transmitter (client) and four as the receivers (APs). The carrier frequency offsets between the WARP transmitter and receivers are measured in the range of several hundreds to several thousands of Hertz. It is much smaller than the sub-carrier size (312.5 kHz) and hence has very little effect on the ToA spectrum. So a carrier frequency offset (CFO) between mobile and AP, and pairs of APs is not a problem for ToneTrack. Each WARP kit is also attached with the FMC-RF-2X245 module to enable four radios on each board as shown in Figure 4.13. The antennas are connected to the WARPs with low loss LMR-400 coaxial cables. All the data recorded at the APs are retrieved through Ethernet connections between the WARPs and the server. My super-resolution MUSIC, spectrum identification (SI), triangle inequality (TI) and clustering schemes are implemented on the server side.

AP Calibration. Due to the nonlinearity of the receiver front end across each sub-carrier, ToneTrack needs to calibrate the channel frequency response in terms of both amplitude and phase. Note that this calibration is a one-time effort for one power-on-off cycle of the WARP. The calibration steps are describe briefly here. First, the radio of the transmitter to the radio of the receiver is connected with an RF cable. Then, the channel frequency response is calculated for each sub-carrier and the phases across each sub-carrier are calibrated into exact linear relationship with the right slope. The slope can be calculated as $2\pi\Delta f t$, where Δf is the sub-carrier size and t is the signal propagation time which can be calculated carefully by measuring the length of the cable attached between the transmitter and the receiver, adding a correction for the small extra path length caused by the splitter and the internal circuitry of the WARP radios. ToneTrack also calibrates the amplitude of the frequency response across each sub-carrier to be equal. After calibration, I'm able to achieve a very sharp time of arrival spectrum close to a line with only one signal transmitted through cables. This front-end linear calibration can be restricted only to ToneTrack processing. The calibration does not factor into the transmit waveform either. The calibration coefficients are calculated as phases and amplitudes for each sub-carrier and they are applied only when ToneTrack processing is called.

4.3 Evaluation

To show how well ToneTrack performs in real indoor environment, I present the results from the testbed described in Section 4.2. First I present my evaluation methodology. Then I show my main results in Section 4.3.2 which answer the following:

1. What is the overall end-to-end performance with channel combination (§4.3.2)?

2. How much is spectrum identification scheme helping ToneTrack (§4.3.2.2)?
3. How does the triangle inequality scheme perform in identifying the APs with direct path totally blocked (§4.3.2.3)?
4. Will increasing numbers of APs improve performance (§4.3.2.4)?
5. What is the performance of ToneTrack with different levels of time synchronization errors between APs (§4.3.2.5)?

After the main results are presented, the choice of W_t is justified in Section 4.3.3.

4.3.1 Experimental methodology

For my experiments, three radios on each AP is utilized to receive signals at channels 1, 5 and 9 respectively. The three radios are connected to a single antenna with combiners. The transmitter either hops across frequencies with one radio, transmitting on three channels sequentially or transmits simultaneously on all three channels with three radios. At each AP position, both data traces from frequency hopping and traces from simultaneous transmissions at multiple channels are collected. They do not have obvious performance difference. My results presented here include all the traces.

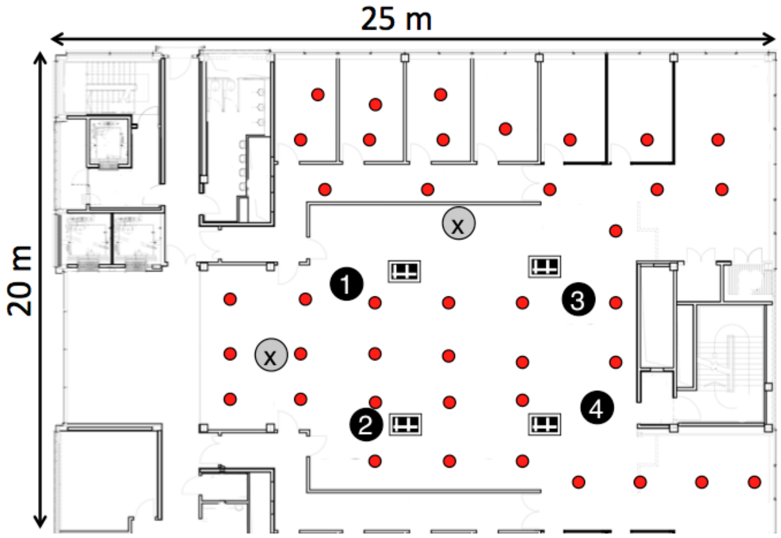


Figure 4.14: The indoor office environment testbed used for my experiments. The four APs used in my experiments are marked as black numbers while the client locations are marked as red dots.

The APs are placed in a 25 × 20 m office, denoting them with numbers shown in Figure 4.14. The clients are placed at 40 randomly-chosen locations denoting their positions as red dots on the floor plan. 12 clients are not in the same room as the APs, with at least one to two walls in between. Please note that only four APs are employed for the evaluation except in Section 4.3.2.4 where the performance of varying number of APs is evaluated.

4.3.2 End-to-end localization accuracy

The end-to-end performance evaluation of ToneTrack is shown in this section.

4.3.2.1 Overall performance

The overall performance of ToneTrack is shown in Figure 4.15. With only three 20 MHz channels, ToneTrack is able to achieve 0.9 m median accuracy in a typical office environment with strong multipaths. The median accuracies of two and one channel are 1.3 m and 1.9 m respectively, significantly better than the naïve resolution. With three channels, the 90% accuracy is around 2 m. The red curve is the CDF plot for super-resolution MUSIC without any of my proposed schemes. So even with just one channel, ToneTrack is able to reduce the median localization error by 40% compared to the state-of–the-art super-resolution scheme. With the channel combination schemes applied, ToneTrack further reduces the median error to below one meter which is a significant improvement with only 20 MHz channels.

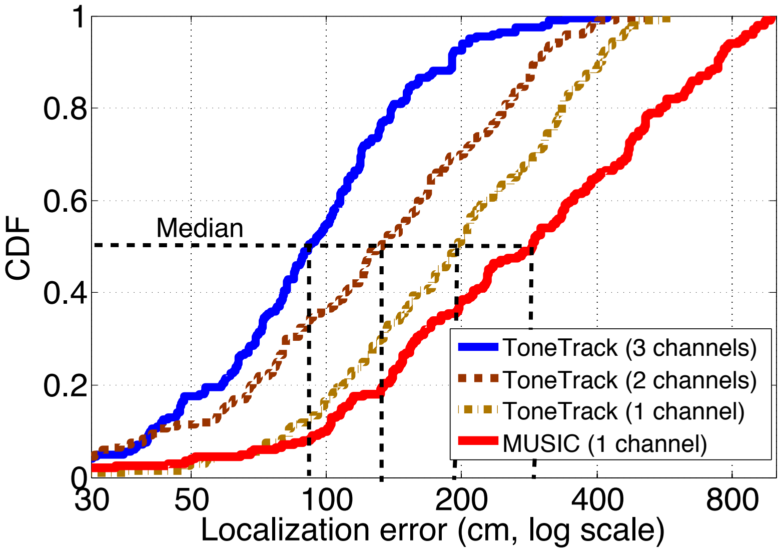


Figure 4.15: ToneTrack's overall localization performance with different numbers of channels and four APs.

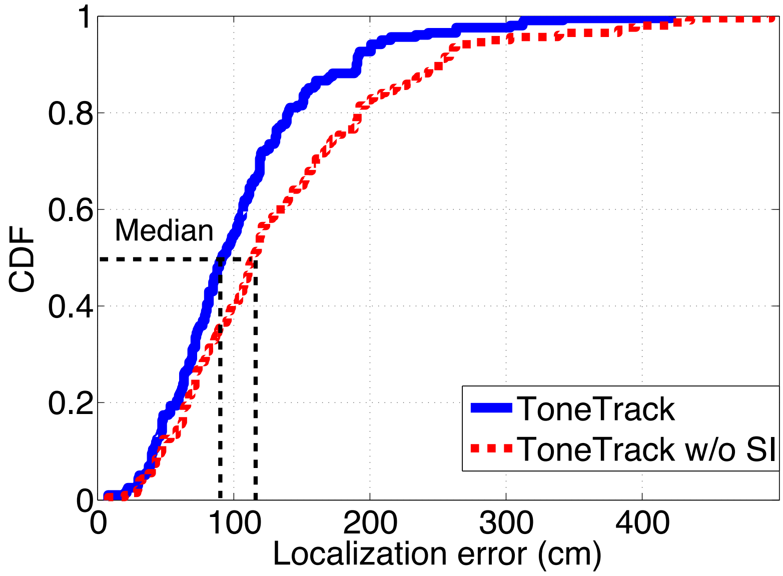


Figure 4.16: Isolating the effect of the spectrum identification (SI) scheme with three channels. Four APs are used in this experiment.

Also the long tail of MUSIC curve is removed in ToneTrack. The effectiveness of channel combination is demonstrate here with three channels in 2.4 GHz band. More channels can be utilized for combination at 5 GHz and 60 GHz bands which means even finer accuracy level can be achieved. I believe with the channel combination scheme proposed in ToneTrack, it is possible to achieve localization accuracy close to UWB systems.

4.3.2.2 Benefit of Spectrum Identification

The effect of spectrum identification (SI) scheme is now isolated and shown in Figure 4.16. With the spectrum identification scheme, the median accuracy is improved from 116 cm to 90 cm. It is clear that the spectrum identification scheme is effective in improving the performance by identifying the more accurate part of the spectrum for localization. However, it is noted that when there are only three APs, this scheme may not be applied because discarding the inaccurate spectrum reduces the number of APs below three which is the minimum requirement for TDoA localization. However, due to the popularity of Wi-Fi in enterprises and universities, this is not an issue as most of the time many APs can be overheard in range. It is also noted that this spectrum identification scheme is more effective in the environment with stronger multipaths which makes it a suitable candidate for indoor localization.

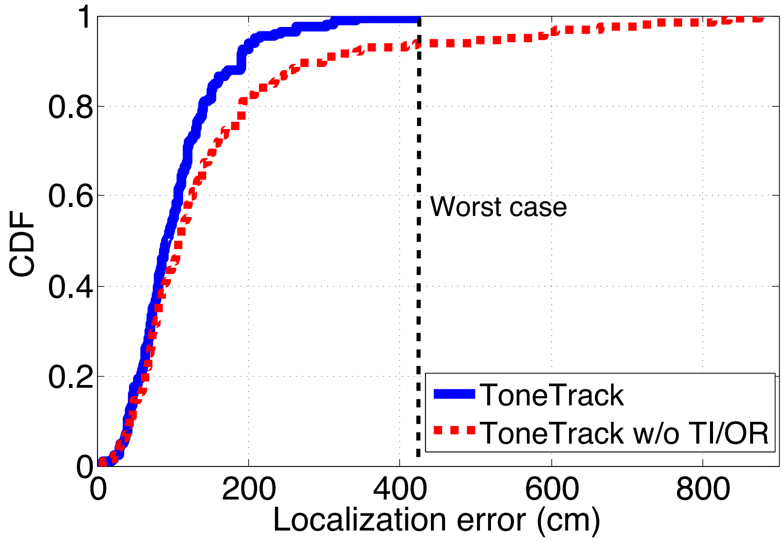


Figure 4.17: The effect of triangle inequality (TI) and clustering schemes.

4.3.2.3 Impact of TI and clustering

Now the triangle inequality (TI) and clustering schemes are removed to see how the performance of ToneTrack is degraded. It is clear from Figure 4.17 that without these schemes, there is a long tail on the CDF. These two schemes are effective in identifying those 'bad' APs (APs with direct path 100% blocked) and estimates with large errors. These 'bad' APs usually cause a big error because only the reflection paths exist and they localize the client to random positions. The direct-path blockage issue is more severe than multipaths. ToneTrack can still try to differentiate the direct path and multipaths if both exist. With direct path 100% blocked, unless ToneTrack can identify the AP and remove it from localization, it always causes a large error which significantly degrades the performance.

4.3.2.4 Number of APs

The effect of varying number of APs on ToneTrack is evaluated in this section with two more APs added at positions marked in Figure 4.14. In order to localize a client, ToneTrack needs a minimum number of three APs to have at least two hyperbolas to intersect. With only three APs, all the schemes proposed are not applied because no single AP can be removed. From the results in Figure 4.18, a clear gap can be seen between the CDF of three APs and four APs. With more APs added, the performance increases slightly. I believe the best solution is to identify the optimal group of APs

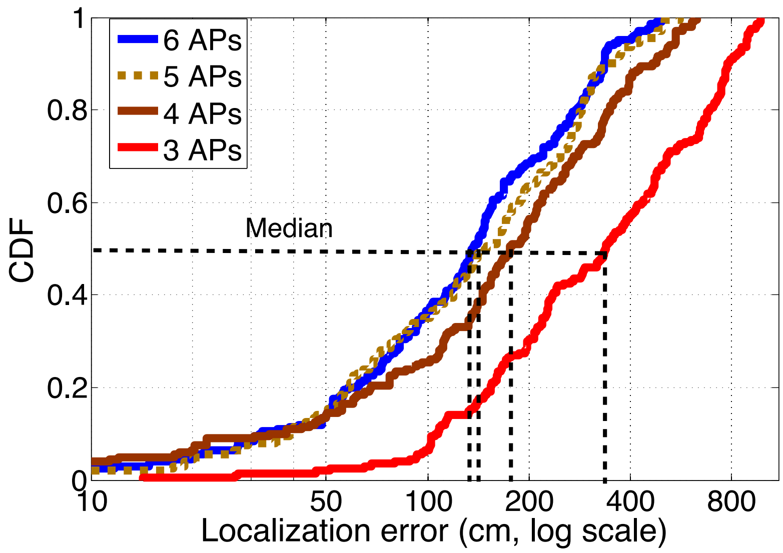


Figure 4.18: ToneTrack's performance with varying number of APs. Only one channel is used in this experiment.

rather than include more random APs for localization. ToneTrack is able to detect the 'bad' APs whose direct path is 100% blocked and remove them. However, it is still challenging to tell which group of APs presents the best localization performance. A safe solution is to include more APs for ToneTrack.

4.3.2.5 Impact of synchronization error

In my current testbed, all the APs are fully synchronized with distributed antenna system deployment. In a distributed MIMO system, there are still time synchronization errors between APs, leading to a performance degradation of ToneTrack. In order to evaluate the performance of ToneTrack with time synchronization error, I borrow the time synchronization error data from SourceSync [80] and incorporate them into my time estimates. Then ToneTrack employs the new TDoA estimates to localize the clients. As shown in Figure 4.19, with 5 ns and 10 ns (95^{th} percentile[4]) time synchronization error, ToneTrack still performs quite well, achieving a median localization accuracy of 1.05 m and 1.4 m respectively with three channels. This time synchronization error is expected to be further reduced in future to have an even less effect on ToneTrack's localization performance.

[4] Note that 5 ns and 10 ns are the 95th percentile values, which mean the average values are significantly smaller.

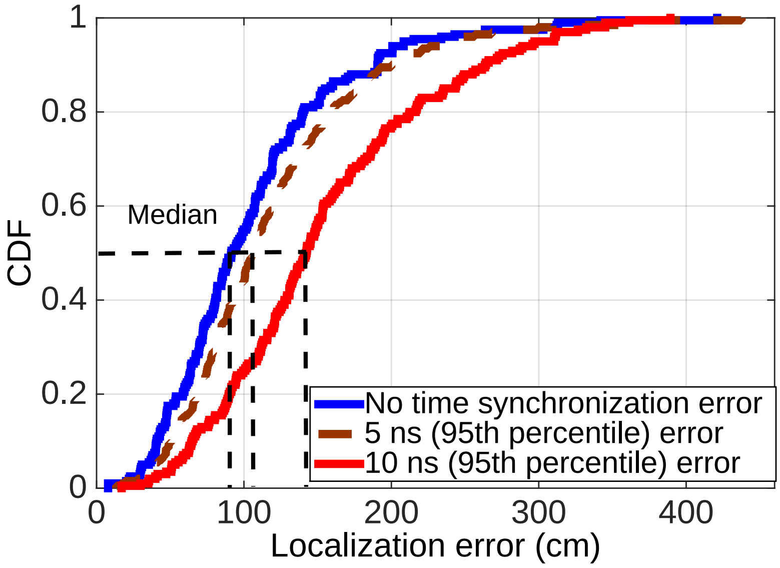


Figure 4.19: ToneTrack's performance with 5 ns and 10 ns (95^{th} percentile) inter-AP time-synchronization error.

4.3.3 Microbenchmark: Choosing W_t

The choice for the spectrum lobe width threshold W_t to differentiate the single peak and merged peak is justified here. When more than two signals are merged or the signals are in the medium and low SNR regions, the width of the merged lobe is much larger. The most challenging scenario is shown in Figure 4.20 where only two signals are merged and they are in the high SNR region (21 dB). With two signals in the high SNR region, the lobe width is the thinnest among the merged lobes. It is shown that even under these conditions, ToneTrack can still choose a constant threshold value safely for a particular bandwidth with very little performance degradation. From Figure 4.20, it can be seen that the width of the merged peak is large as long as the path difference between the two signals are above 1.7 m. If ToneTrack chooses the threshold as 2.5 m[5] to differentiate a single and merged peak, ToneTrack makes mistakes only when the path difference of the two signals is below 1.5 m. Note that the merged peak position is always between the true peak positions of the two signals. When the path difference is as small as below 1.5 m, the deviation of the merged peak position from the true direct path peak position is also small. So mis-identification of the merged peak as single peak in this scenario has little effect in the performance. From Figure 4.21, it is clear that the lobe width of

[5] Note that the spectrum lobe width is converted from time at the speed of light.

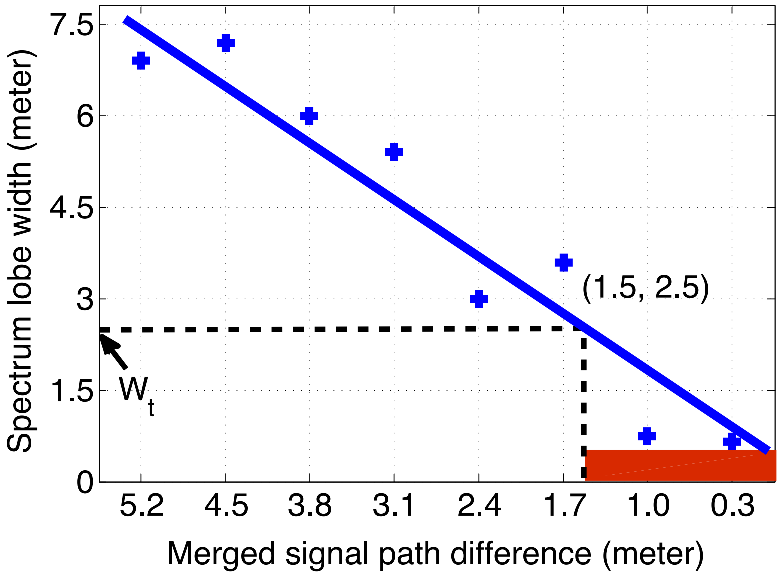


Figure 4.20: The merged peak width decreases when the signal path difference decreases (21 dB SNR).

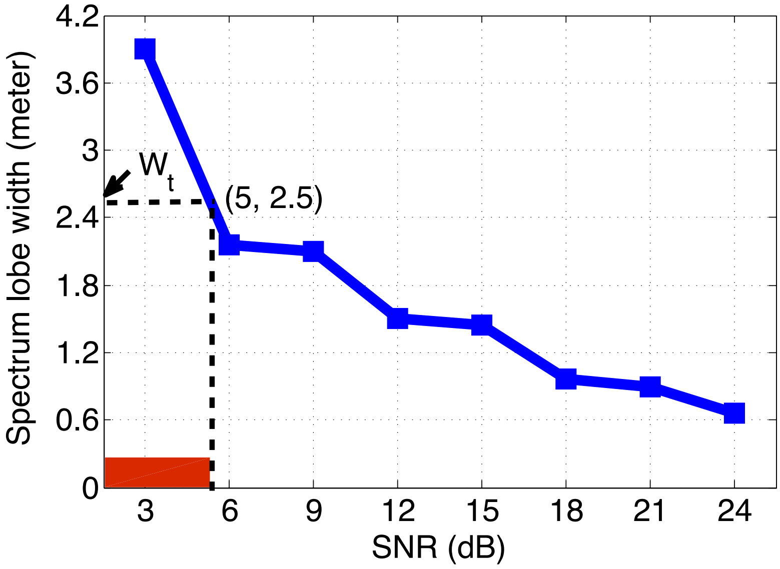


Figure 4.21: The lobe width of a single signal's ToA spectrum decreases when SNR increases. The lobe width increases dramatically when SNR goes below 6 dB. The red region denotes a range where ToneTrack classifies a single signal peak as a merged peak, but note the extremely low SNR.

a single signal remains well below 2.5 m as long as the SNR is above 6 dB. ToneTrack makes mistakes only in the very low SNR region (below 6 dB). In this SNR region, the accuracy level of the spectrum is anyway low and ToneTrack relies more on other APs to localize the client. So the effect of making mistakes on an inaccurate spectrum is also small. It is also noted that with many APs around, it is unlikely all the APs have low SNRs with respect to a client. This threshold can be adjusted with varying SNR values in the future work. However, a single threshold is performing pretty well as explained.

Chapter 5

Future Work

The current ArrayTrack is a 2-D localization system. Several immediate extensions of ArrayTrack are listed as below:

3-D tracking: It is straightforward to extend ArrayTrack to 3-D with another dimension of antenna array added. To include another linear array, nearly twice the number of antennas as current are required.

Location with massive MIMO: With the 60 GHz frequency band introduced in the next generation Wi-Fi and cellular standards, large number of antennas will be attached to the AP and the accuracy level can be significantly increased.

Combine ArrayTrack with other schemes: ArrayTrack is flexible to be combined with other localization schemes for better performance. ArrayTrack can easily be applied on top of ToneTrack to further refine the location estimates.

ToneTrack is able to achieve a 0.9 m median accuracy with three adjacent 20 MHz channels. It is expected that the accuracy level will be significantly increased with the following future plans:

Wider bandwidth: With the latest 802.11ac standard, the channel bandwidth supported can be up to 160 MHz as shown in Figure 5.1. Evaluating ToneTrack on higher bandwidth channels is the next step to boost the performance. The third generation WARP platform only supports a maximum bandwidth of 36 MHz, which is limited by the MAX2829 transceiver used. Employing a UWB platform and down-sampling the time domain data to the bandwidth required is one option to obtain data from larger bandwidth channels currently not available.

Data combination from non-adjacent channels: Adjacent channel combination has been successfully realized in ToneTrack. In indoor environment, because of strong

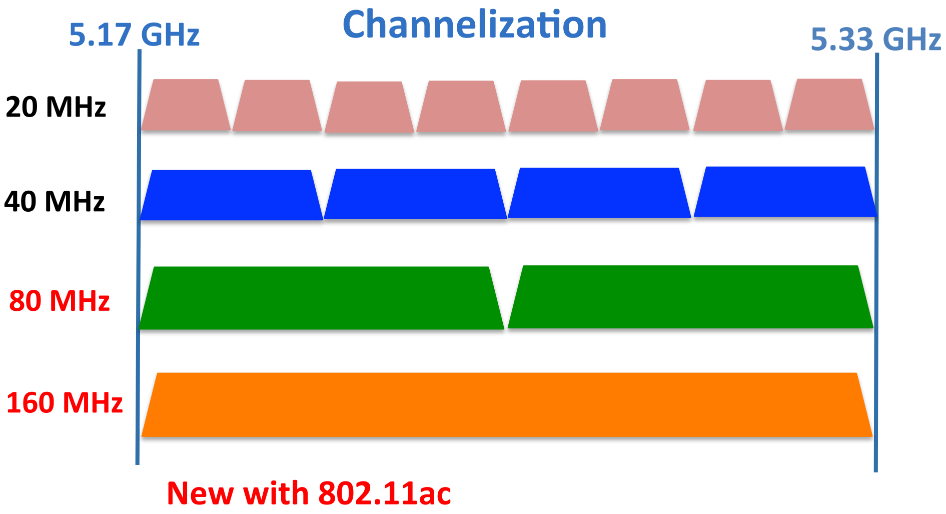


Figure 5.1: 802.11ac supports up to 160MHz bandwidth

multipath, the phase information across each sub-carrier is not linear. This non-linear relationship makes phase offset calculation critical for channel combination extremely difficult. Combining non-adjacent channels for localization is a very interesting direction worth exploring. A further extension of non-adjacent channel combination is to combine data from different carrier frequency bands such as 2.4, 5 and 60 GHz.

Wi-Fi imaging: With several 160 MHz channels combined, ToneTrack is able to achieve millimeter level accuracy. With such a high level of accuracy, Wi-Fi imaging may become possible which will enable many new applications!

Chapter 6

Conclusion

In this thesis, I have presented two systems I built during my Ph.D period to push the state-of-the-art indoor localization limits.

I have presented the design, implementation and evaluation of ArrayTrack, the first indoor location system hosted on Wi-Fi Infrastructure which is able to achieve submeter accuracy with AoA scheme. The most challenging part of AoA localization is the presence of strong multipath reflections indoors, which leads to severe performance degradation. I have proposed a novel multipath identification scheme to handle this problem. ArrayTrack is robust in term of signal to noise ratio, collision and mobile orientation. ArrayTrack does not require any offline training and the computational load is small, making it a great candidate for real time location services. With six 8-antenna APs, ArrayTrack is able to achieve a median accuracy below 30 cm.

The second system I have presented is a TDoA based indoor localization system called ToneTrack. I have proposed a innovative channel combination scheme to increase the effective bandwidth which significantly improves the accuracy. I have proposed a novel spectrum identification scheme to retrieve useful information even when the overall ToA spectrum is mostly inaccurate. The triangle inequality and outlier rejection schemes are used to handle the challenging scenario when the LOS path is 100% blocked. ToneTrack is able to achieve below one meter accuracy with only three transmissions from adjacent 20 MHz channels. With the scheme proposed in ToneTrack, it is possible to combine more channels in the 5 GHz band to achieve localization accuracy close to UWB systems.

Thus ArrayTrack and ToneTrack push the envelope of localization systems in terms of accuracy, hardware requirements, and responsiveness.

Bibliography

[1] Rice argos project: Practical many-antenna base stations. http://argos.rice.edu.

[2] A Amar and A Weiss. Fundamental resolution limits of closely spaced random signals. *IET Radar and Sonar Navigation*, 2(3):170–179, 2008.

[3] E Aryafar, N Anand, T Salonidis, and EW Knightly. Design and experimental evaluation of multi-user beamforming in wireless LANs. In *Proceedings of ACM MobiCom*, 2010.

[4] Distributed Antenna System: Keeping customers connected, wherever they are (AT&T). `http://www.att.com/gen/press-room?pid=23351`.

[5] P Bahl, J Padhye, L Ravindranath, M Singh, A Wolman, and B Zill. DAIR: A framework for managing enterprise wireless networks using desktop infrastructure. In *Proceedings of ACM HotNets*, 2005.

[6] P Bahl and V Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of IEEE INFOCOM*, pages 775–784, 2000.

[7] P Bahl, V Padmanabhan, and A Balachandran. Enhancements to the RADAR location tracking system. Technical Report MSR-TR-2000-12, Microsoft Research, February 2000.

[8] HV Balan, R Rogalin, A Michaloliakos, K Psounis, and G Caire. Airsync: Enabling distributed multiuser mimo with full spatial multiplexing. *ACM Transactions on Networking*, 21(6), 2013.

[9] CA Boano, N Tsiftes, T Voigt, J Brown, and U Roedig. The impact of temperature on outdoor industrial sensornet applications. *IEEE Transactions on Industrial Informatics*, 6(3):451–459, 2010.

[10] S Bratus, C Cornelius, D Kotz, and D Peebles. Active behavioral fingerprinting of wireless devices. In *Proceedings of ACM WiSec*, pages 56–61, March 2008.

[11] Broadcom Inc. White Paper: Ethernet time synchronization. `http://www.broadcom.com/collateral/wp/StrataXGSIV-WP100-R.pdf`.

[12] S Capkun, M Hamdi, and J Hubaux. GPS-free positioning in mobile ad-hoc networks. In *Proceedings of Hawaii Int'l Conference on System Sciences*, 2001.

[13] P Castro, P Chiu, T Kremenek, and R Muntz. A probabilistic room location service for wireless networked environments. In *Proceedings of Ubiquitous Computing*, 2001.

[14] Y Chan and K Ho. A simple and efficient estimator for hyperbolic location. *IEEE Trans. on Sig. Proceedings*, 42(8), 1994.

[15] R Chandra, R Mahajan, T Moscibroda, R Raghavendra, and P Bahl. A case for adapting channel width in wireless networks. In *Proceedings of ACM SIGCOMM*, 2008.

[16] SA Chelloug. Impact of the temperature and humidity variations on link quality of xm1000 mote sensors. *arXiv preprint arXiv:1501.01073*, 2015.

[17] H Chen, T Lin, H Kung, and Y Gwon. Determining RF angle of arrival using COTS antenna arrays: a field evaluation. In *Proceedings of the MILCOM Conf.*, 2012.

[18] Y Chen, D Lymberopoulos, J Liu, and B Priyantha. FM-based indoor localization. In *Proceedings of ACM MobiSys*, 2012.

[19] K Chintalapudi, A Iyer, and V Padmanabhan. Indoor localization without the pain. In *Proceedings of ACM MobiCom*, 2010.

[20] K Chintalapudi, B Radunovic, V Balan, M Buettener, S Yerramalli, V Navda, and R Ramjee. WiFi-NC: Wi-Fi over narrow channels. In *Proceedings of USENIX NSDI*, 2012.

[21] Y Chu and A Ganz. A uwb-based 3d location system for indoor environments. In *IEEE 2nd International Conference on Broadband Networks*, 2005.

[22] Positioning Statement on Cisco Wireless LAN over Distributed Antenna Systems (Cisco Corp.).

[23] L Cong and W Zhuang. Hybrid TDoA/AoA mobile user location for wideband CDMA cellular systems. *IEEE Trans. on Wireless Communications*, 1(3):439–447, 2002.

[24] D Dardari, A Conti, U Ferner, A Giorgetti, and M Win. Ranging with ultrawide bandwidth signals in multipath environments. *Proceedings of the IEEE*, 97(2):404–426, February 2009.

[25] J Do, M Rabinowitz, and P Enge. Performance of TOA and TDOA in a nonhomogeneous transmitter network combining GPS and terrestrial signals. In *Proceedings ION National Technical Meeting*, 2006.

[26] J Elson, L Girod, and D Estrin. Fine-grained network time synchronization using reference broadcasts. *ACM SIGOPS Operating Systems Review*, 36(SI):147–163, 2002.

[27] D Faria and D Cheriton. No long-term secrets: Location based security in overprovisioned wireless LANs. In *Proceedings of the* ACM *HotNets Workshop*, 2004.

[28] J Fessler and A Hero O. Space-alternating generalized expectation-maximization algorithm. *IEEE Transactions on Signal Processing*, 42(10):2664–2677, 1994.

[29] GJ Foschini and MJ Gans. On limits of wireless communications in a fading environment when using multiple antennas. *Wireless Personal Communications*, 6(3):311–335, March 1998.

[30] D Gesbert, M Kountouris, RW Heath Jr, CB Chae, and T Sälzer. Shifting the mimo paradigm. *Signal Processing Magazine, IEEE*, 24(5):36–46, 2007.

[31] J Gjengset, J Xiong, G McPhilips, and K Jamieson. Phaser: Enabling phased array signal processing on commodity Wi-Fi access points. In *Proceedings of ACM MobiCom*, 2014.

[32] SA Golden and SS Bateman. Sensor Measurements for Wi-Fi Location with Emphasis on Time-of-Arrival Ranging. *IEEE Transactions on Mobile Computing*, 6(10), 2007.

[33] Connected vehicles. `http://www.massdotinnovation.com/Pdfs/Session%202B-Connected%20Vehicles.pdf`.

[34] D Gore, RW Heath Jr, AJ Paulraj, et al. Transmit selection in spatial multiplexing systems. *Communications Letters, IEEE*, 6(11):491–493, 2002.

[35] Y Gwon and R Jain. Error characteristics and calibration-free techniques for wireless LAN-based location estimation. In *ACM MobiWac*, 2004.

[36] A Haeberlen, E Flannery, A Ladd, A Rudys, D Wallach, and L Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *Proceedings of ACM MobiCom*, 2004.

[37] R Harle and A Hopper. Deploying and evaluating a location-aware system. In *Proceedings of ACM MobiSys*, 2005.

[38] P Hu, L Li, C Peng, G Shen, and F Zhao. Pharos: Enable physical analytics through visible light based indoor localization. In *Proceedings of ACM HotNets*, 2013.

[39] IEEE Standard 802.11b-1999: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band, September 1999. `http://standards.ieee.org/getieee802/802.11.html`.

[40] IEEE Standard 802.11a-2003: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 5 GHz Band, June 2003. `http://standards.ieee.org/getieee802/802.11.html`.

[41] IEEE Standard 802.11g-2003: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band, June 2003. `http://standards.ieee.org/getieee802/802.11.html`.

[42] IEEE Standard 802.11n-2009: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 5: Enhancements for Higher Throughput, October 2009. `http://standards.ieee.org/getieee802/802.11.html`.

[43] IEEE 802.11ac Specification Driven by Evolving Market Need For Higher, Multi-User Throughput In Wireless Lans, January 2014. `http://standards.ieee.org/getieee802/802.11.html`.

[44] SJ Ingram, D Harmer, and M Quinlan. Ultrawideband indoor positioning systems and their use in emergencies. In *Position Location and Navigation Symposium, 2004*, pages 706–715. IEEE, 2004.

[45] Intel Ultimate N WiFi Link 5300. `http://www.intel.com/products/wireless/adapters/5000/`.

[46] Z Jiang, J Zhao, X Li, J Han, and W Xi. Rejecting the Attack: Source Authentication for Wi-Fi Management Frames using CSI Information. In *Proceedings of IEEE INFOCOM*, 2013.

[47] K Joshi, S Hong, and S Katti. PinPoint: Localizing Interfering Radios. In *Proceedings of NSDI*, 2013.

[48] P Krishnan, A Krishnakumar, W Ju, C Mallows, and S Ganu. A system for LEASE: Location estimation assisted by stationary emitters for indoor RF wireless networks. In *Proceedings of IEEE INFOCOM*, 2004.

[49] S Krishnan, P Sharma, G Zhang, and O Woon. A uwb based localization system for indoor robot navigation. In *Ultra-Wideband, 2007. ICUWB 2007. IEEE International Conference on*, pages 77–82. IEEE, 2007.

[50] S Kumar, S Gil, D Katabi, and D Rus. Accurate indoor localization with zero start-up cost. In *Proceedings of ACM MobiCom*, 2014.

[51] S Kumar, E Hamed, D Katabi, and L E Li. LTE radio analytics made easy and accessible. In *Proceedings of ACM SIGCOMM*, 2014.

[52] Y-S Kuo, P Pannuto, K-J Hsiao, and P Dutta. Luxapose: Indoor positioning with mobile phones and visible light. In *Proceedings of ACM MobiCom*, 2014.

[53] T.I Laakso, V Valimaki, M Karjalainen, and U.K Laine. Splitting the unit delay. *IEEE Sig. Proceedings Mag.*, 13(1):30–60, 1996.

[54] A Ladd, K Bekris, A Rudys, G Marceau, L Kavraki, and D Wallach. Robotics-based location sensing using wireless ethernet. In *Proceedings of ACM MobiCom*, 2002.

[55] V Lang and C Gu. A locating method for WLAN based location service. In *IEEE International Conference on e-Business Engineering*, 2005.

[56] L Li, P Hu, C Peng, G Shen, and F Zhao. Epsilon: A visible light based positioning system. In *Proceedings of USENIX NSDI*, 2014.

[57] L Li, G Shen, C Zhao, T Moscibroda, J-H Lin, and F Zhao. Experiencing and handling the diversity in data density and environmental locality in an indoor positioning service. In *Proceedings of ACM MobiCom*, 2014.

[58] T. Li, A. Ekpenyong, and Y.-F. Huang. A location system using asynchronous distributed sensors. In *Proceedings of IEEE INFOCOM*, 2004.

[59] X Li and K Pahlavan. Super-resolution TOA estimation with diversity for indoor geolocation. *IEEE Trans. on Wireless Comms.*, 3(1), 2004.

[60] Z Li, W Chen, C Li, M Li, X Li, and Y Liu. Flight: Clock calibration using fluorescent lighting. In *Proceedings of ACM MobiCom*, 2012.

[61] Z Li, W Dehaene, and G Gielen. A 3-tier uwb-based indoor localization system for ultra-low-power sensor networks. *Wireless Communications, IEEE Transactions on*, 8(6):2813–2818, 2009.

[62] H Lim, C Kung, J Hou, and H Luo. Zero configuration robust indoor localization: Theory and experimentation. In *Proceedings of IEEE INFOCOM*, 2006.

[63] H Liu, Y Gan, J Yang, S Sidhom, Y Wang, Y Chen, and F Ye. Push the limit of Wi-Fi based localization for smartphones. In *Proceedings of ACM MobiCom*, 2012.

[64] K Liu, X Liu, and X Li. Guoguo: Enabling fine-grained indoor localization via smartphone. In *Proceedings of ACM MobiSys*, 2013.

[65] D C Loh, C Y Cho, C P Tan, and R S Lee. Identifying unique devices through wireless fingerprinting. In *Proceedings of the* ACM *WiSec Conf.*, pages 46–55, March 2008.

[66] D Madigan, E Elnahrawy, R.P Martin, W Ju, P Krishnan, and A.S Krishnakumar. Bayesian indoor positioning systems. In *Proceedings of IEEE INFOCOM*, 2005.

[67] S Marano, W Gifford, H Wymeersch, and M Win. Nlos identification and mitigation for localization based on uwb experimental data. *Selected Areas in Communications, IEEE Journal on*, 28(7):1026–1035, 2010.

[68] A Mariakakis, S Sen, J Lee, and K-H Kim. SAIL: Single access point-based indoor localization. In *Proceedings of ACM MobiSys*, 2014.

[69] Dual-Band 802.11a/b/g World-Band MAX2829 Transceiver. `http://www.maximintegrated.com/en/datasheet/index.mvp/id/4532`.

[70] G Mellen et al. Closed-form solution for determining emitter location using time difference of arrival measurements. *IEEE Trans. on Aerospace and Electronic Systems*, 39(3), 2003.

[71] LS Monteiro, T Moore, and C Hill. What is the accuracy of dgps? *Journal of Navigation*, 58(02):207–225, 2005.

[72] R Nandakumar, K Chintalapudi, and V Padmanabhan. Centaur: Locating devices in an office environment. In *Proceedings of ACM MobiCom*, 2012.

[73] D Niculescu and B Nath. Ad-hoc positioning system (APS) using AoA. In *Proceedings of IEEE INFOCOM*, 2003.

[74] D Niculescu and B Nath. VOR base stations for indoor 802.11 positioning. In *Proceedings of ACM MobiCom*, 2004.

[75] S Nirjon, J Liu, G DeJean, B Priyantha, Y Jin, and T Hart. COIN-GPS: Indoor localization from direct GPS receiving. In *Proceedings of ACM MobiSys*, 2014.

[76] N Patwari and S Kasera. Robust location distinction using temporal link signatures. In *Proceedings of ACM MobiCom*, pages 111–122, September 2007.

[77] AJ Paulraj and T Kailath. US 5345599 A: Increasing capacity in wireless broadcast systems using distributed transmission/directional reception (DTDR), 1993. Stanford Junior University.

[78] N Priyantha, H Balakrishnan, E Demaine, and S Teller. Mobile-assisted localization in wireless sensor networks. In *Proceedings of IEEE INFOCOM*, 2005.

[79] N Priyantha, A Chakraborty, and H Balakrishnan. The Cricket location-support system. In *Proceedings of ACM MobiCom*, pages 32–43, August 2000.

[80] H Rahul, H Hassanieh, and D Katabi. SourceSync: A distributed wireless architecture for exploiting sender diversity. In *Proceedings of ACM SIGCOMM*, 2010.

[81] H Rahul, S Kumar, and D Katabi. MegaMIMO: Scaling Wireless Capacity with User Demands. In *ACM SIGCOMM 2012*, Helsinki, Finland, August 2012.

[82] A Rai, K Chintalapudi, V Padmanabhan, and R Sen. Zee: Zero-effort crowdsourcing for indoor localization. In *Proceedings of ACM MobiCom*, 2012.

[83] GG Raleigh and JM Cioffi. Spatio-temporal coding for wireless communications. In *Proceedings GLOBECOM*, 1996.

[84] T S Rappaport. *Wireless Communications: Principles and Practice*. Prentice-Hall, 2nd edition, 2002.

[85] Rice Univ. Wireless Open Access Research Platform (WARP). `http://warp.rice.edu/trac`.

[86] T Roos, P Myllymaki, and H Tirri. A probabilistic approach to WLAN user location estimation. *International J. of Wireless Information Networks*, 9(3), 2002.

[87] R Roy and T Kailath. ESPRIT—Estimation of signal parameters via rotational invariance techniques. *IEEE Transaction on Acoustics, Speech, and and Signal Processing*, 37(7):984–995, July 1989.

[88] A Saleh, A Rustako, and R Roman. Distributed antennas for indoor radio communications. IEEE *Trans. on Comms.*, COM-35(12):1245–51, 1987.

[89] T Sarkar and O Pereira. Using the matrix pencil method to estimate the parameters of a sum of complex exponentials. *IEEE Antenna and Propagation Magazine*, 37(1), 1995.

[90] A Savvides, C Han, and M Srivastava. Fine-grained localization in ad-hoc networks of sensors. In *Proceedings of ACM MobiCom*, 2001.

[91] T M Schmidl and D C Cox. Robust Frequency and Timing Synchroniation for OFDM. *IEEE Transaction on Communications*, 45(12):1613–1621, December 1997.

[92] R Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Transaction on Antennas and Propagation*, AP-34(3):276–280, March 1986.

[93] R Schmidt. Least squares range difference location. *IEEE Trans. on Aerospace and Electronic Systems*, 32(1):234–242, 1996.

[94] S Sen, R Choudhury, and S Nelakuditi. SpinLoc: Spin once to know your location. In *HotMobile*, 2012.

[95] S Sen, J Lee, K Kim, and P Congdon. Avoiding multipath to revive inbuilding WiFi localization. In *Proceedings of ACM MobiSys*, 2013.

[96] S Sen, B Radunovic, R Choudhury, and T Minka. Spot localization using PHY layer information. In *Proceedings of ACM MobiSys*, 2012.

[97] T-J Shan, M Wax, and T Kailath. On spatial smoothing for direction-of-arrival estimation of coherent signals. *IEEE Transaction on Acoustics, Speech, and Signal Processing*, ASSP-33(4):806–811, August 1985.

[98] A Sheth, S Seshan, and D Wetherall. Geo-fencing: Confining Wi-Fi Coverage to Physical Boundaries. In *Proceedings of the 7th International Conference on Pervasive Computing*, 2009.

[99] A Smailagic, D Siewiorek, J Anhalt, D Kogan, and Y Wang. Location sensing and privacy in a context aware computing environment. In *Pervasive Computing*, 2001.

[100] R Steele. *Mobile Radio Communications*. IEEE Press, 1994.

[101] K Tan, J Fang, Y Zhang, S Chen, L Shi, J Zhang, and Y Zhang. Fine-grained channel access in wireless LAN. In *Proceedings of ACM SIGCOMM*, 2010.

[102] K Tan, H Liu, J Fang, W Wang, J Zhang, M Chen, and G Voelker. SAM: Enabling practical spatial multiple access in wireless LAN. In *Proceedings of ACM MobiCom*, 2009.

[103] K Tan, J Zhang, J Fang, H Liu, Y Ye, S Wang, Y Zhang, H Wu, W Wang, and G Voelker. Sora: High performance software radio using general purpose multi-core processors. In *Proceedings of USENIX NSDI*, April 2009.

[104] A Tarighat, N Khajehnouri, and A Sayed. Improved wireless location accuracy using antenna arrays and interference cancellation. 4, 2003.

[105] Time Domain, Inc. `http://www.timedomain.com`.

[106] BD Van Veen and KM Buckley. Beamforming: A versatile approach to spatial filtering. *IEEE assp magazine*, 5(2):4–24, 1988.

[107] M Vanderveen, B Ng, C Papadias, and A Paulraj. Joint angle and delay estimation (JADE) for signals in multipath environments. In *Asilomar Conf. on Signals, Systems, and Computers*, 1996.

[108] M Vanderveen, C Papadias, and A Paulraj. Joint angle and delay estimation (JADE) for multipath signals arriving at an antenna array. *IEEE Communications L.*, 1(1):12–14, January 1997.

[109] A Varshavsky, E Lara, J Hightower, A LaMarca, and V Otsason. GSM indoor localization. In *Pervasive and Mobile Computing*, 2007.

[110] H Wang, S Sen, A Elgohary, M Farid, M Youssef, and R Choudhury. No need to war drive: Unsupervised indoor localization. In *Proceedings of ACM MobiSys*, 2012.

[111] J Wang, F Adib, R Knepper, D Katabi, and D Rus. RF-Compass: Robot object manipulation using RFIDs. In *Proceedings of ACM MobiCom*, 2013.

[112] J Wang and D Katabi. Dude, where's my card? RFID positioning that works with multipath and non-line of sight. In SIGCOMM, 2013.

[113] J Wang, D Vasisht, and D Katabi. RF-IDraw: Virtual touch screen in the air using RF signals. In *Proceedings of ACM SIGCOMM*, 2014.

[114] Y Wang and L Song. An algorithmic and systematic approach for improving robustness of ToA-based localization. In *IEEE Conf. on High Performance Computing and Communications, Embedded, and Ubiquitous Computing*, 2013.

[115] R Want, A Hopper, V Falcao, and J Gibbons. The active badge location system. *ACM Trans. on Information Systems*, 10(1):91–102, January 1992.

[116] A Ward, A Jones, and A Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4(5):42–47, October 1997.

[117] C Wong, R Klukas, and G Messier. Using WLAN infrastructure for angle-of-arrival indoor user location. In *Proceedings of the IEEE VTC Conf.*, pages 1–5, September 2008.

[118] H Xia, A Herrera, S Kim, and F Rico. A CDMA-distributed antenna system for in-building personal communications services. IEEE JSAC, 1996.

[119] Y Xie, Z Li, and M Li. Precise power delay profiling with commodity Wi-Fi. In *Proceedings of ACM MobiCom*, 2015.

[120] Y Xie, Y Wang, P Zhu, and X You. Grid-search-based hybrd ToA/AoA location techniques for NLOS environments. *IEEE Comms. Letters*, 13(4):254–256, 2009.

[121] Xilinx FPGA (Virtex-4). `http://www.xilinx.com`.

[122] J Xiong and K Jamieson. Secureangle: Improving wireless security using angle-of-arrival signatures. In *Proceedings of ACM HotNets*, 2010.

[123] J Xiong and K Jamieson. Towards fine-grained radio-based indoor location. In *Proceedings of ACM HotMobile*, 2012.

[124] J Xiong and K Jamieson. ArrayTrack: A Fine-Grained Indoor Location System. In *Proceedings of USENIX NSDI*, 2013.

[125] J Xiong and K Jamieson. SecureArray: Improving WiFi security with fine-grained physical-layer information. In *ACM MobiCom*, 2013.

[126] J Xiong, K Jamieson, and K Sundaresan. Synchronicity: Pushing the envelope of fine-grained localization with Distributed MIMO. In *HotWireless*, 2014.

[127] J Xiong, K Sundaresan, K Jamieson, M Khojastepour, and S Rangarajan. Midas: Empowering 802.11ac networks with multiple-input distributed antenna systems. In *Proceedings of ACM CoNEXT*, 2014.

[128] L Xiong. A selective model to suppress nlos signals in angle-of-arrival (AoA) location esimation. In *Proceedings of the IEEE PIMRC*, 1998.

[129] Xirrus Corp. (`http://www.xirrus.com`).

[130] L Yang, Y Chen, XY Li, C Xiao, M Li, and Y Liu. Tagoram: Real-time tracking of mobile RFID tags to high precision using COTS devices. In *Proceedings of ACM MobiCom*, 2014.

[131] Z Yang, Y Liu, and X Li. Beyond trilateration: On the localizability of wireless ad-hoc networks. In *Proceedings of IEEE INFOCOM*, 2009.

[132] Z Yang, C Wu, and Y Liu. Locating in fingerprint space: Wireless indoor localization with little human intervention. In *Proceedings of ACM MobiCom*, 2012.

[133] S Yoon, K Lee, and I Rhee. FM-based indoor localization via automatic fingerprint DB construction and matching. In *Proceedings of ACM MobiSys*, 2013.

[134] M Youssef and A Agrawala. The Horus WLAN location determination system. In *Proceedings of ACM MobiSys*, 2005.

[135] Y Zhou, C Look Law, Y Guan, and F Chin. Indoor elliptical localization based on asynchronous uwb range measurement. *Instrumentation and Measurement, IEEE Transactions on*, 60(1):248–257, 2011.